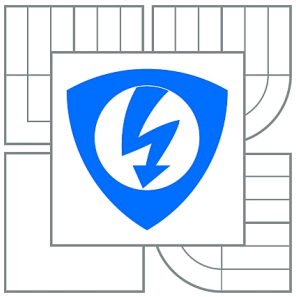
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS

TOPOLOGIE SÍTÍ A JEJICH MONITOROVÁNÍ

NETWORK TOPOLOGIES AND THEIR MONITORING

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

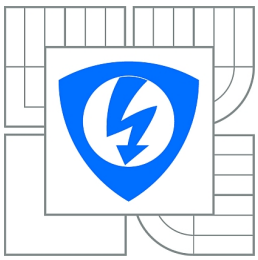
AUTOR PRÁCE — Bc. MIROSLAV SIROTNÝ

AUTHOR

VEDOUCÍ PRÁCE — Ing. MICHAL POLÍVKA

SUPERVISOR

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student:	Bc. Miroslav Sirotný	***ID:***	77873
Ročník:	2	***Akademický rok:***	2009/2010

NÁZEV TÉMATU:

Topologie sítí a jejich monitorování

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte problematiku počítačových sítí. Zaměřte se na aktuálně používané topologie. Specializujte se na 3. a 7. vrstvu ISO OSI modelu, konkrétně na protokoly HTTP, ICMP. Navrhněte intervence do TCP/IP sítě vedoucí k narušení jej funkčnosti. Navrhněte a prakticky realizujte robota určeného k průzkumu a mapování topologie počítačové sítě, libovolné velikosti. Předpokládá se, že robot bude pracovat s protokoly HTTP a ICMP. Robot bude analyzovat zejména propustnost jednotlivých segmentů sítě a topologii. Analyzujte možnosti útoku na infrastrukturu sítě, při zohlednění její topologie, získané jejím mapováním vytvořeným robotem.

DOPORUČENÁ LITERATURA:

[1] BARABÁSI, Albert-László. V pavučině sítí. Věra Amelová; RNDr. František Slanina Csc. 1. vyd. Praha: Ladislav Horáček - Paseka, 2005. 280 s. Fénix; sv. 13. ISBN 80-7185-751-3
[2] SCAMBRAY, Joel, MCCLURE, Stuart, KURTZ, George. Hacking bez tajemství. Praha: Computer Press, 2001. 592 s. ISBN 80-7226-549-0.
[3] ENDORF, Carl, SCHULTZ, Eugene, MELLANDER, Jim. Hacking - Detekce a prevence počítačového útoku. Praha: Grada, 2005. 356 s. ISBN 80-247-1035-8.
[4] DOSTÁLEK, Libor, KABELOVÁ, Alena. Velký průvodce protokoly TCP/IP a systémem DNS. 3. vyd. Praha: Computer Press, 2002. 542 s. ISBN 80-7226-675-6.
[5] SHINDER, Debra Littlejohn. Počítačové sítě., Softpress, 2003. 752 s. ISBN 8086497550

Termín zadání:	29.1.2010	***Termín odevzdání:***	26.5.2010

Vedoucí práce: Ing. Michal Polívka

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Diplomová práca pojednáva o počítačových sieťach, ktoré tvoria v súčasnej dobe globálnu komunikačnú štruktúru a zohrávajú v dnešnej spoločnosti veľmi dôležitú úlohu. Možno si to väčšina z nás ani neuvedomuje ale prichádzame s nimi do styku skoro neustále. Umožňujú nám komunikovať prostredníctvom Internetu napríklad pomocou služieb: email, skype, icq, facebook, apod.

Práca je zameraná na návrh a realizáciu robota určeného k prieskumu a mapovaniu topológie počítačovej siete a to na protokole HTTP a ICMP. V teoretickej časti sa venujeme počítačovým sieťam, siedmej a tretej vrstve ISO/OSI modelu, protokolom HTTP a ICMP, útokom na DNS, DoS útokom a systémom detekcie a prevencie narušenia.

KLÚČOVÉ SLOVÁ

Počítačová sieť, HTTP, ICMP, DNS, Útok, Jáva, Detekcia, Prevencia

ABSTRACT

The master’s thesis deals with computer networks, which are currently the global communication infrastructure and play a very important role in today’s society. Most of us can be unaware of how often we interact with these networks. We almost constantly come into contact with them. They allow us to communicate through the Internet via services such as: email, skype, icq, facebook, etc...

The work focuses on the design and implementation of a robot designed for exploration and mapping of computer network topology and protocols HTTP and ICMP. The theoretical part is dedicated to computer networks, seventh and third-layer ISO/OSI model, protocol HTTP and ICMP attacks against DNS, DoS attacks and detection systems and avoiding distortaions.

KEYWORDS

Computer Network, HTTP, ICMP, DNS, Attack, Java, Detection, Prevention

SIROTNÝ, M. *Topologie sítí a jejich monitorování. Brno*: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 53 strán, 2 prílohy. Vedúci diplomovej práce Ing. Michal Polívka.

Prehlásenie

Prehlasujem, že svoju diplomovú prácu na tému *Topologie sítí a jejich monitorování* som vypracoval samostatne pod vedením vedúceho diplomovej práce a s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej diplomovej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb, taktiež som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných alebo majetkových a som si plne vedomí následkov porušení ustanovení § 11 a nasledujúceho autorského zákona č. 121/2000 Sb., v rámci možných trestne právnych dôsledkov vyplývajúcich z ustanovení § 152 trestného zákona č. 140/1961 Sb.

V Brne dňa ....

(podpis autora)

Poďakovanie

Ďakujem vedúcemu diplomovej práce Ing. Michalovi Polívkovi za veľmi užitočnú, metodickú, pedagogickú a odbornú pomoc a ďalšie cenné rady pri spracovaní diplomovej práce.

V Brne dňa ....

(podpis autora)

Obsah

Úvod **12**

1. Definícia a rozdelenie počítačových sietí **13**
1.1 Podľa rozlohy sa počítačové siete delia na: 13
1.1.1 Lokálna počítačová sieť – LAN (Local Area Network) 13
1.1.2 Metropolitná sieť – MAN (Metropolitan Area Network) 13
1.1.3 Rozľahlá sieť – WAN (Wide Area Network) 13
1.1.4 Sieť PAN (Personal Area Network) 13
1.2 Podľa topológie sa počítačové siete delia na: 14
1.2.1 Fyzická topológia 14
1.2.2 Logická topológia 16

2. Referenčný ISO/OSI model **17**
2.1 Aplikačná vrstva 17
2.1.1 Protokol HTTP (Hypertext Transport Protocol) 18
2.1.2 Verzie HTTP protokolu 18
2.1.3 Fungovanie protokolu 18
2.1.4 Metódy prokolu HTTP 19
2.1.5 Proxy server 19
2.2 Sieťová vrstva 21
2.2.1 Protokol ICMP (Internet Control Message Protocol) 22
2.2.2 Zprávy ICMP paketu 23
2.2.3 Echo 23
2.2.4 Nedoručený IP datagram (Destination unreachable) 26
2.2.5 Zníž rýchlosť odosielania (Source quench) 26
2.2.6 Zmeň smerovanie (Redirect) 26
2.2.7 Žiadosť o smerovanie 26
2.2.8 Čas vypršal (Time exceeded) 27
2.2.9 Požiadavka o masku (Address Mask Request) 29
2.2.10 Časová synchronizácia (Time synchronization) 29

3. Útoky na DNS a systémy smerovania paketov **30**
3.1 Systém DNS (Domain Name System) 30
3.2 Útoky ohrozujúce smerovanie paketov 31
3.3 Útoky na DNS servery 31
3.4 Útok typu DNS spoofing (falšovanie DNS údajov) 31
3.5 Útok typu DNS poisoning (otrava DNS) 32

4. Návrh a praktická realizácia robota **34**
4.1 Programovací jazyk Java a vývojové prostredie Netbeans IDE 6.8 34
4.3 Návrh a realizácia robota pracujúceho s protokolom HTTP 34
4.4 Návrch a realizácia robota pracujúceho s protokolmi HTTP a ICMP 40

5. Priepustnosť (Throughput) siete **43**
5.1 Techniky pre zvýšenie priepustnosti siete: segmentácia 43

6. DoS (Denial of Service) útoky **44**
6.1 Záplavové DoS útoky (DoS Flood) 45

6.1.1 ICMP záplava (ICMP Flood) 45
6.1.2 Smurf Attack 45
6.1.3 TCP záplavy (TCP Flood) 45
6.1.4 UDP záplavy (UDP Flood) 45

7. Systémy prevencie a narušenia (IPS a IDS) 46
7.1 IPS a IDS systémy 46
7.2 Systémy prevencie narušenia (IPS – Intrusion Prevention System) 46
7.3 Systémy detekcie narušenia (IDS – Intrusion Detection System) 47

8. Záverečné zhodnotenie 48

Zoznam skratiek 51

Zoznam príloh 53

Zoznam obrázkov

Obr. 1: Zbernicová a kruhová topológia 15
Obr. 2: Hviezdicová topológia a rozšírená hviezda 15
Obr. 3: Stromová a zmiešaná topológia 15
Obr. 5: Priebeh komunikácie medzi klientom a serverom 18
Obr. 6: Fungovanie komunikácie sprostredkované proxy serverom [4] 20
Obr. 7: Celková štruktúra ICMP paketu 22
Obr. 8: Štruktúra ICMP hlavičky 22
Obr. 9: Štruktúra a zprávy ICMP paketu [6] 23
Obr. 10: Zistenie dostupnosti uzlu 85.248.69.111 24
Obr. 11: Obsah správy typu ICMP echo request zachytenej vo Wireshraku 25
Obr. 12: Obsah správy typu ICMP Echo reply zachytenej vo Wireshraku 25
Obr. 13: Obsah správy ICMP Echo request 25
Obr. 14: Obsah správy ICMP Echo reply 25
Obr. 15: Zmeň smerovanie [8] 26
Obr. 16: Príkaz tracert 27
Obr. 17: Príklad ICMP paketu 28
Obr. 18: Ukážka programu tracert v prostredí Windows 28
Obr. 19: Časová synchronizácia [6] 29
Obr. 20: Zistenie IP adresy zo sieťového rozhrania www.sme.sk 30
Obr. 21: Spôsob komunikácie v DNS systéme [4] 30
Obr. 22: Infiltrácia do DNS cache [10] 31
Obr. 23: Útok typu DNS spoofing 32
Obr. 24: Komunikácia medzi klientom a DNS serverom 33
Obr. 25: Útok typu DNS poisoning 33
Obr. 26: Vytvorené menu programu 34
Obr. 27: Metóda SetupTracert slúži na vybratie sieťového rozhrania 35
Obr. 28: Fungovanie metódy Run 35
Obr. 29: Ukážka zakomentovaného bloku 36
Obr. 27: Rozdiel medzi absolútnou a relatívnou adresou 37
Obr. 30: Príklad fungovania metódy FindLinksOnPage 38
Obr. 32: Množina PrehladanychLinkov 39
Obr. 31: Množina NaPrehladanie 39
Obr. 33: Ochrana proti zacykleniu 39
Obr. 34: Fungovanie metódy Stop 39
Obr. 35: Ukladanie zachytených odkazov do súboru zachytené_odkazy.txt 40
Obr. 36: Vytvorené menu programu 40
Obr. 37: Vytvorený robot pracujúci s protokolmi HTTP a ICMP 41
Obr. 38: Ukážka stránky, na ktorú nebude vykonaný program tracert 41
Obr. 39: Ukážka nedostupnosi uzla (smerovača) 42
Obr. 40: Ukladanie výstupu programu do súboru tracert.txt 42
Obr. 41: Techniky pre zvýšenie priepustnosti siete [17] 43
Obr. 42: Útok typu DDoS [19] 44
Obr. 43: Systém prevencie narušenia IPS 46
Obr. 44: Systém prevencie narušenia IPS 46
Obr. 43: Systém detekcie narušenia IDS 47

Zoznam tabuliek

Tab. 1: Definície topológií, ich výhody a nevýhody 14

Úvod

Obdobie 21. storočia môžeme charakterizovať aj ako storočie rýchleho rozvoja informačno-komunikačných technológií. Každý mesiac sú na trh uvádzané nové, modernejšie a výkonnejšie produkty pre spracovanie informácií. Moderné technológie sú súčasťou nášho každodenného života. Internet a počítačové siete jednoznačne patria k nim. Internet nám ponúka dostupné, rýchle a aktuálne informácie z ktorejkoľvek oblasti života. Prudký rozvoj počítačovej a komunikačnej techniky spôsobil, že dnešnú spoločnosť môžeme nazvať informačnou. Takmer každý z nás dennodenne či už v práci, škole alebo vo voľnom čase využíva Internet a počítačové siete.

Získané teoretické poznatky o počítačových sieťach, o 7. a 3. vrstve ISO/OSI modelu, o protokoloch HTTP a ICMP, o útokoch na DNS, o DoS útokoch a systémoch prevencie a detekcie narušenia sú popísané v teoretickej časti diplomovej práce.

Praktická časť práce je rozdelená na dve časti: prvá časť je venovaná návrhu a praktickej realizácií robota určeného k priezkumu počítačovej siete ľubovoľnej veľkosti na aplikačnej vrstve a to na protokole HTTP. Vytvorený robot dokáže na webovej stránky odkaz: vyhľadať, spracovať, vybrať a vypísať. Druhá časť je venovaná návrhu a praktickej realizácií robota určeného k priezkumu a mapovaniu topológie počítačovej siete ľubovoľnej veľkosti na aplikačnej a sieťovej vrstve a to na protokole HTTP a ICMP. Pri vytvárený tejto aplikácie je využitý predchádzajúci program, ktorý slúži na zachytávanie odkazov na stránke a program tracert, ktorý vypisuje uzly na ceste paketov od zdrojového počítača k cieľovému počítaču. Vytvorený robot slúži tzv. mapovaniu ľubovoľnej počítačovej siete. Vytvorená aplikácia je realizovaná v programovacom jazyku Jáva s vývojovým prostredím NetBeans 6.8.

1. Definícia a rozdelenie počítačových sietí

Počítačová sieť je systém vzájomne prepojených a spolupracujúcich počítačov. Medzi týmito počítačmi možno prostredníctvom siete pohodlne a rýchlo prenášať informácie [1].

1.1 Podľa rozlohy sa počítačové siete delia na:

1.1.1 Lokálna počítačová sieť – LAN (Local Area Network)

Sú to dátové siete, ktoré prepájajú jednotlivé počítače a servery na malom geografickom území – sú to stovky metrov, možno aj pár kilometrov. V praxi si to ide predstaviť ako počítačovú sieť v rámci jednej budovy alebo viacej budov, ktoré sú blízko seba. Prenosové rýchlosti sa pohybujú v rozmedzí 10 až 1000 Mbit/s [2].

1.1.2 Metropolitná sieť – MAN (Metropolitan Area Network)

Jedná sa o sieť, ktorá navzájom spája lokálne (LAN) siete na menšom geografickom území napr. medzi viacerými budovami až v rozsahu mesta, do jednej siete. Prenosové rýchlosti sú v rozmedzí 100 Mbit/s až 10 Gbit/s [2].

1.1.3 Rozľahlá sieť – WAN (Wide Area Network)

Táto sieť dokáže spájať menšie počítačové siete typu LAN a MAN po celom svete. Jedná sa teda o verejné siete, ktoré sú svojím rozsahom neobmedzené, napr. zaberajú územie štátov a kontinentov. Sú do nich pripojené tisíce až milióny počítačov. Používajú sa ku komunikácií užívateľov a k získavaniu informácií. Prenosové rýchlosti sa tu pohybujú v rozmedzí 100 Mbit/s až 1 Gbit/s prípadne aj vyššími. V týchto sieťach sa len výnimočne využíva technológia ethernet ale tiež aj všetky druhy prenosových médií - metalické, optické, bezdrôtové rádiové [2].

1.1.4 Sieť PAN (Personal Area Network)

Jedná sa o bezdátovú sieť s veľmi krátkym dosahom (do 10 metrov). Tieto siete sa využívajú na prepojenie zariadení, akými sú napríklad klávesnice, tlačiarne, ktoré využíva užívateľ (zákazník) každý deň pri práci s počítačom. Pri tvorený PAN sietí sa používajú bezdrôtové technológie WiFi, IrDA (Infrared Data Association), Bluetooth.

Špecifické postavenie má sieť „Internet“ – je to počítačová sieť, ktorá vznikla prepojením rôznych LAN, MAN, WAN sietí.

1.2 Podľa topológie sa počítačové siete delia na:

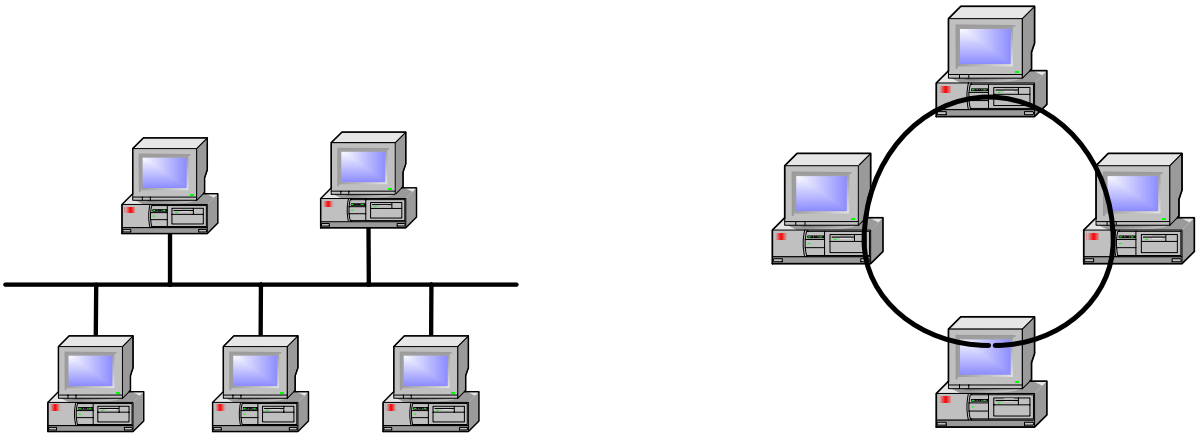
Medzi základné typy topológií sietí LAN zaraďujeme napríklad zbernicu , kruh a hviezdu. Ďalej poznáme aj zložitejšie topológie sietí a tie získame kombináciou základných typov, najčastejšia kombinácia hviezd do stromovej štruktúry.

1.2.1 Fyzická topológia

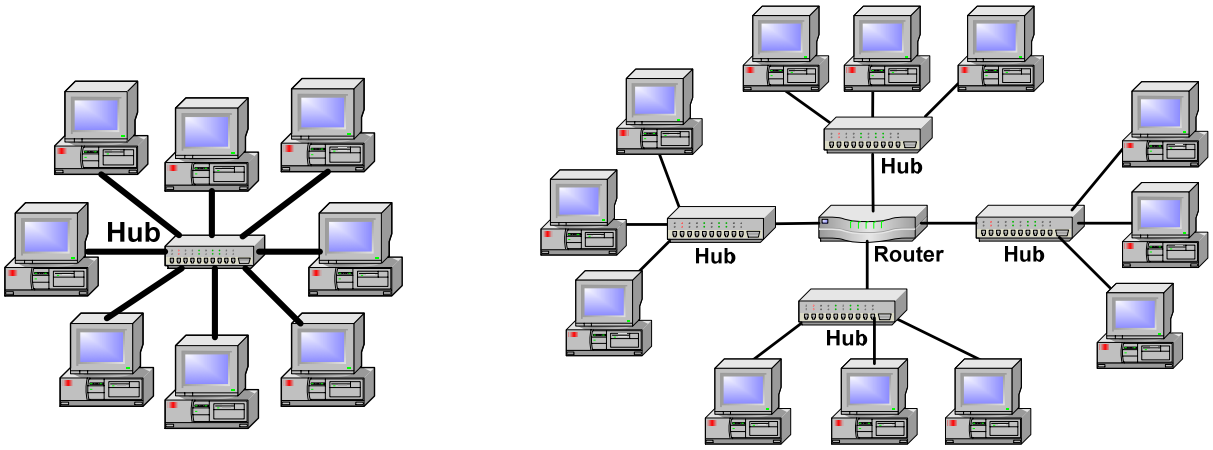
Porovnanie jednotlivých tologogií je znázornené v tab. 1.

Zbernicová (bus) Všetky uzly sú pripojené k jednému priamemu vedeniu a žiadne iné spojenie medzi nimi neexistuje obr. 1. **Výhody** – Všetky zariadenia sú navzájom prepojené, spolu komunikujú, jednoduché pripojenie, lacné riešenie. **Nevýhody** – Pri porušení prenosového média je sieť nefunkčná, malá prenosová rýchlosť.
Kruhová (ring) Topológiu sietí určujú uzly, ktoré tvoria jednoduchý uzavretý kruh. Každý uzol je spojený s dvomi susednými uzlami. Všetky zariadenie sú navzájom priamo prepojené. Správa sa šíri jedným smerom od jedného k druhému uzlu a každé uzol posiela správu ďalej k svojmu susedovi, až pokiaľ nedôjde k cieľovému uzlu obr. 1. **Výhody** – Jednoduché pripojenie, lacné riešenie. **Nevýhody** – Zlyhanie jedného uzlu ma dopad na celú sieť, veľmi málo sa používa v paterných sieťach.
Hviezdicová (star) V topológií je centrálny uzol, ku ktorému sa pripájajú všetky ďalšie uzly, obvykle je to prepínač alebo rozbočovač. V topológií sa nachádza centrálne zariadenie, cez ktoré prechádzajú dáta. Správa sa šíri po celom vedeni a cieľová stanica ju prijíma obr. 2. **Výhody** – Ľahká modifikácia a pridávanie nových uzlov, zlyhanie jedného uzla, neovplyvní fungovanie siete, používa sa v paterných sieťach. **Nevýhody** – Ak zlyhá centrálny uzol zruší sa celá sieť.
Rozšírená hviezda (extended star) Každý uzol je stredom ďalšej hviezdy obr. 2. **Výhody** – V rozšírenej hviezde sa limituje počet zariadení pripojených do jedného centra.
Stromová (tree) Stromovú typológiu je časť rozšírenej hviezdy, s kmeňom stromu, miesto centra obr. 3. **Výhody** – Jednoduché rozširovanie siete, používa sa v paterných sieťach. **Nevýhody** – Pri výpadku centrálneho uzla je nefunkčný celý podstrom siete.
Zmiešaná (mesh) V zmiešanej topologii sú jednotlivé uzly vzájomne prepojené obr. 3. **Výhody** – Ak niektorá z vetiev vypadne, tak sa sieť neporuší. **Nevýhody** – Je vhodná pri použití malých sietí (LAN).

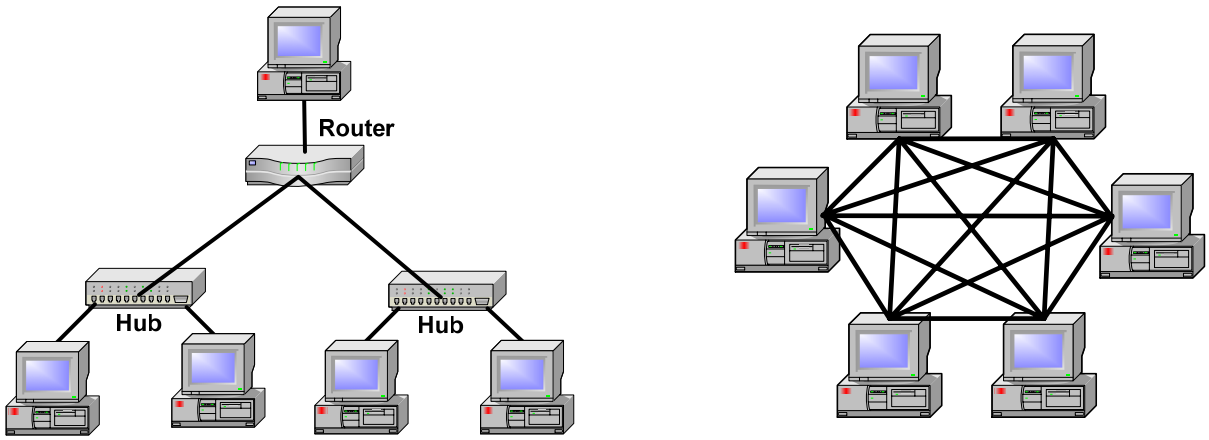
Tab. 1: Definície topológií, ich výhody a nevýhody



Obr. 1: Zbernicová a kruhová topológia



Obr. 2: Hviezdicová topológia a rozšírená hviezda



Obr. 3: Stromová a zmiešaná topológia

V ďalšom texte je vysvetlené čo znamenajú pojmy: rozbočovač (hub), smerovač (router) a prepínač (switch).

- **Rozbočovač (Hub)** – Rozbočovač a prepínač sú zariadenia, ktoré pracujú s dátami usporiadanými do dátových rámcov. Rámce, ktoré prijímu zosilnia a pošlú na port cieľového počítača. Hlavný rozdiel medzi týmito zariadeniami je v tom, ako posielajú rámce k cieľovému počítaču. Každý jeden dátový rámec je určený pre konkrétny počítač v sieti. Rozbočovač je zariadenia, ktoré nevie na aký port má daný dátový rámec poslať, tak ho posiela na všetky porty. Týmto vie zabezpečiť, že rámec sa dostane na príslušné počítače, ale zbytočne zaťažuje komunikáciu v sieti tým, že rámce sa rozpošlú na všetky počítače a len ten pravý ho spracuje, ostatné počítače ho ignorujú [3].

- **Smerovač (Router)** – Smerovač je zariadenia, ktoré spracováva dátové pakety. Jeho hlavnou úlohou je smerovať pakety do inej počítačovej siete. Paket neobsahuje len dáta, ale aj cieľovú adresu, na ktorú má byť doručený. Smerovač väčšinou prepája dve alebo viac sietí (LAN, WAN a podobne). Pomocou hlavičky paketu dokáže smerovač určiť najlepšiu cestu pre jeho doručenie [3].

- **Prepínač (Switch)** – Prepínač je zariadenie, ktoré obsahuje vnútornú pamäť. V tejto pamäti si uchováva sieťové adresy (MAC – Media Access Control => tzv. je to hardvérová adresa, ktorá jednoznačne identifikuje každé zariadenie v sieti) pripojených počítačov. Ak prepínač prijime dátový rámec, presne vie, na ktorom porte je pripojený počítač, ktorému je rámec určený a vysiela ho len na tento port. To veľmi urýchľuje komunikáciu v sieti a naviac prepínač môže pre komunikáciu s pripojeným počítačom využiť celú šírku komunikačného pásma [3].

1.2.2 Logická topológia

Táto topológia siete nám naznačuje, akým spôsobom sa môžu dáta v sieti šíriť medzi stanicami. Logická topológia je nezávislá od fyzickej. V praxi využívame dve logické topológie:

- **Zbernica** – Táto logická topológia sa používa na šírenie dát. Tieto dáta sa šíria naraz od vysielajúcej stanice k všetkým ostatným staniciam. Hlavným predstaviteľom tejto topológie je Ethernet, v ktorej sa dodnes využíva zbernicová logická topológia, nezávisle na konkrétnej fyzickej topológií inštalovanej ethernetovskej siete. S nástupom prepínačov (switchov) sa tok dát v sieťach zracionalizoval (inak povedané: rámce nie sú posielané všetkým staniciam, ale len do tej časti siete, v ktorej sa nachádza príjemca), avšak povahu zbernicovej topológie to nemení.

- **Kruh** – V logickej topológií sa doručované dáta posúvajú v istom pevnom poradí od jednej stanice k druhej. Tá stanica, ktorej dáta prináležia, si ich prevezme a všetky ostatné stanice tieto dáta ignorujú.

2. Referenčný ISO/OSI model

Referenčný model ISO/OSI na obr. 4 je najznámejší vrstvový model popisujúci sieťovú architektúru. Jedná sa o sedemvrstvový hierarchický model .

Obr. 4: Referenčný model ISO/OSI

2.1 Aplikačná vrstva

V referenčnom modeli ISO/OSI sa jedná o najvyššiu vrstvu. Je to vrstva zodpovedná za poskytovanie prístupu aplikáciám do siete. Pomocou aplikačnej vrstvy môžeme prevádzať napr. prenos súborov (FTP), elektronickú poštu (SMTP), prístup k webovým stránkam (HTTP), apod. Medzi protokoly a programy, ktoré poskytujú služby aplikačnej vrstvy patria napríklad:

- **FTP** (File Transfer Protocol) – FTP pracuje na princípe klien – server. Tento protokol slúži sa pre prenos súborov medzi vzdialenými počítačmi. Využíva porty TCP/20 a TCP/21.

- **HTTPS** (Hypertext Transport Protocol Secure) – HTTPS je nádstavba protokolu HTTP, ktorá poskytuje bezpečnost pred odpočúvaním. HTTPS protokol komunikuje na porte 443. Bezpečnosť HTTTS závisí na implementácií ako na serveru tak aj na klientovi.

- **SMTP** (Simple Mail Transfer Protocol) – SMTP je protokol pomocou ktorého klient odosiela správy a pomocou ktorého sa tieto správy prenášajú medzi servery. Poskytuje službu elektronickej pošty. Tento protokol používa TCP pre prijímanie a odosielanie správ elektronickej pošty. SMTP funguje nad protokolom TCP a používa port TCP/25.

- **POP3** (Post Office Protocol verzie 3) – Tento protokol pracuje na princípe klient – server a používa sa k sťahovaniu emailových správ.

2.1.1 Protokol HTTP (Hypertext Transport Protocol)

Protokol HTTP je aplikačný protokol, ktorý je určený pôvodne pre výmenu hypertextových dokumentov vo formáte HTML. Slúži ku komunikácií medzi klientom a WWW serverom. HTTP protokol definuje tvar dát, ktoré sú prenášané a tiež aj formát dotazov a odpovedí komunikujúcich strán. Používa obvykle port TCP/80 [4].

2.1.2 Verzie HTTP protokolu

- **HTTP/0.9** (Hypertext Transfer Protocol verzie 0.9) – Táto verzia zaisťovala iba prenos dát po Internetu bez ďalších doplňkových informácií o prenášaných dátach. Klient musel odhadnúť podľa koncovky súboru o aké dáta sa jedná.

- **HTTP/1.0** (Hypertext Transfer Protocol verzie 1.0) – Ukázalo sa, že táto varianta nestačí a preto natúpila verzia HTTP protokolu 1.0. Táto verzia sa snažila doplniť popisujúce informácie do požiadavkov a odpovedí a preto použila už existujúci formát MIME. Rozšířila tak tvar požiadavkov a odpovedí o štandardizované doplňujúce informácie charakterizujúce prenášané dáta v tvare typ/podtyp. Verzia protokolu HTTP/1.0 je definovaná v RFC 1945.

- **HTTP/1.1** (Hypertext Transfer Protocol verzie 1.1) – S rozvojom služby WWW (World Wide Web) sa objavili ďalšie požadavky na HTTP protokol. Jednalo sa predovšetkým o požiadavok o trvalé spojenie medzi klientom a serverom. Tieto nedostatky rieši práve nová verzia HTTP protokolu 1.1. Táto verzia zavádza možnosť trvalého spojenia. V rámci tohoto spojenia klient posiela všetky svoje požadavky na server a server mu po tomto spojení posiela svoje odpovede. Vo verzií 1.1 je trvalé spojenie chápané tak, že klient predpokladá, že server udržuje trvalé spojenie a naopak, spojenie zostáva otvorené do tej doby pokiaľ klient alebo server spojenie neukončí. Verzia protokolu HTTP/1.1 je definovaná v RFC 2068.

2.1.3 Fungovanie protokolu

Protokol funguje spôsobom „požiadavka – odpoveď“ viz obr. 5. Proces prebieha tak, že klient, najčastejšie webový prehliadač, pošle požiadavku na danú webovú stránku, protokol HTTP špecifikuje typ správ, ktoré klient odošle serveru, a rovnako aj typ správ, ktoré server odošle klientovi. HTTP protokol je bezestavový protokol, pretože nevie uchovávať samotné data. Protokol HTTP používa rozšírenie tzv. HTTP cookies. Toto rozšírenie umožňuje serveru uchovávať informácie o stave spojenia na počítači [4].

Obr. 5: Priebeh komunikácie medzi klientom a serverom

2.1.4 Metódy prokolu HTTP

Metoda určuje druh služby, ktorú klient od serveru vyžaduje.

OPTIONS – Táto metóda predstavuje dotaz na možnosti komunikácie spojené s uvedeným URL. Metóda umožňuje klientovi určiť možnosti komunikácie so zdrojom alebo schopnosťami serveru, napríklad ak je URL v dotazu v tvare "*", potom sa jedná o dotaz na možnosti serveru ako celku [5].

GET – Metóda GET predstavuje požiadavok, ktorým klient požaduje od serveru dáta. na posielanie dokumentu určeného pomocou URL. V súvislosti s proxy sa môže metóda GET zmeniť na "podmienený GET", ktorá posielať dokument iba za určitých podmienok definovaných hlavičke dotazu [5].

HEAD – Táto metóda je podobná ako metóda GET akurát rozdielom je v tom, že server nemusí posielať telo odpovedí. Používa sa tiež k získaniu doplnkových informácií o dokumente a využíva sa aj k testovaniu hypertextových odkazov a ich dostupnosti.

POST, PUT – PUT a POST sú správy, ktorými sa na webový server umiestňujú jednotlivé dokumenty. Rozdiel je v tom, že správa POST sa využíva v prípade, ak odosielame dáta napríklad z formulára na webovej stránke a pomocou správy PUT je na server umiestňovaný obsah webovej stránky.

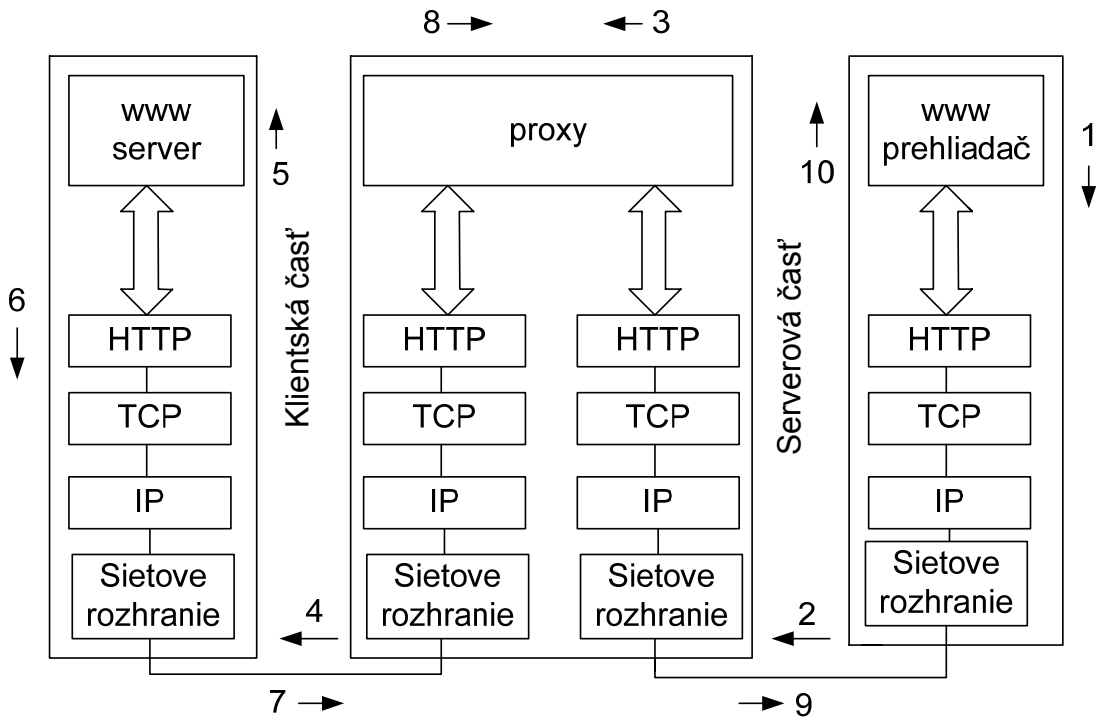
DELETE – Požiadavok na zrušenie dokumentu na serveru. Rušený dokument je špecifikovaný v URL.

TRACE – Metóda TRACE sa používa k testovaniu originálneho serveru, ktorý vráti klientovi kladnú odpoveď bez dát“.

Dnešné WWW servery podporujú metódy GET, POST a HEAD.

2.1.5 Proxy server

Jedná sa o program, ktorý pracuje súčasne ako klient (uživateľ) i server. Proxy server pracuje ako prostredník medzi klientom a cieľovým počítačom. Na obr. 16 znázornené fungovanie komunikácie sprostredkované proxy serverom. Na obrázku je vynechané DNS zisťovanie IP adresy hostiteľa, ktorá prebieha ako na strane proxy serveru tak na strane klienta. Hlavná vlastnosť proxy serveru je taká, že môže vystupovať v roli uživateľa a tým poskytuje užívateľovi anonymitu vzhľadom k WWW sereveru [4].



Obr. 6: Fungovanie komunikácie sprostredkované proxy serverom [4]

Popis jednotlivých krokov je vysvetlený v nasledujúcom texte:

1. Klient si chce prezerať (alebo zobrazovať) stránku na Internete, napr. http://www.vutbr.cz/stranka.html, tak táto úloha sa predá nižším vrstvám protokolu.

2. Tieto vrstvy slúžia k naviazaniu spojenia so serverovou časťou proxy serveru a následne predajú požiadavok prehliadaču (HTTP metóda GET).

3. Proxy server rozumie HTTP protokolu (jeho štruktúre) a základe týchto znalostí dokáže vyhodnotiť požiadavok. Ďalej proxy server dokáže tento požiadavok prepísať tak napr. zmeniť cieľový server, z ktorého sa bude stránka získavať, ďalej dokáže overiť či je uživateľ oprávnení takýto požiadavok previesť.

4. V štvrtom kroku klientská časť zaistí naviazanie spojenia s WWW serverom a predanie metódy (GET/stránka.html).

5. V piatom kroku WWW server prijíma požiadavok a vyhodnocuje ho.

6. Výsledok je odpoveď WWW serveru, obvykle to je požadovaná stránka, ktorú klient očakáva.

7. Na základe znalostí nižších vrstiev je odpoveď predaná klientskej časti proxy serveru.

8. Krok je podobný ako tretí krok, len obrátene napr. prepísanie odpovede do formy v akej ju uživateľ očakáva.

9. V deviatom kroku sa odpoveď predá pomocou nižších vrstev klientovi.

10. Prehliadač stránku spracuje a následne ju zobrazí. V našom prípade je to stránka: http://www.vutbr.cz/stranka.html

2.2 Sieťová vrstva

V referenčnom modeli ISO/OSI sa jedná o tretiu vrstvu. Úlohou tejto vrstvy je zabezpečiť smerovanie dát. Základnou jednotkou prenosu je sieťový paket, ktorý sa balí do dátového rámca a skladá sa zo záhlavia a dátového poľa.

V rozsiahlych sieťach medzi počítačmi obvykle sa nachádza jeden alebo viac smerovačov. Medzi susednými smerovačmi je na linkovej vrstve vždy priame spojenie. Smerovač vybalí z dátového rámca (jedného linkového protokolu) sieťový paket a ešte pred odosielaním do inej linky ho opäť zabalí do jednoho dátového rámca. Ako prepojovací uzol na tejto vrstve môže byť použitý smerovač. Medzi protokoly sieťovej vrstvy patria napríklad:

- **IP** (Internet Protocol) – Jedná sa o najznámejší protokol sieťovej vrstvy. Je to dátový protokol, ktorý je používaný pre prenos dát cez paketové siete. Tvorí základný protokol dnešného Internetu. IP protokol prenáša tzv. „IP datagramy“ medzi vzdialenými počítačmi. Každý IP datagram nesie vo svojom záhlaví adresu príjemcu, čo je úplná smerovacia informácia pre dopravu IP datagramov k príjemcovi. Každý IP datagram sa teda môže prenášať samostatne. IP datagramy teda k príjemcovi nemusia vôbec doraziť, môžu byť duručené viackrát a môžu doraziť v inom poradí ako boli odosielané.

- **IPv4** (Internet Protocol verzie 4) – IPv4 je paketovo orientovaný protokol, ktorý funguje spôsobom „best effort“ (best effort => nespolahlivá služba). Tento protokol poskytuje unikátný identifikátor (IP adresu) a zaisťuje jednoznačnú identifikáciu stanice v rámci Internetu. Zaisťuje tiež smerovania dát k požadovanému cieľu.a teda aj možnosť smerovania dát prenášaných sieťou k danému cieľu. Protokol IPv4 používa 32 bitové adresy [4].

- **IPv6** (Internet Protocol verzie 6) – Podstata protokolu IPv6 je rovnaká ako pri IPv4. Možnosť smerovania dát prenášaných sieťou k danému cieľu. Protokol IPv6 používa 128 bitové adresy [4].

- **IGMP** (Internet Group Management Protocol) – Tento protokol sa používa k riadeniu multicastových skupín, napr. pridanie (alebo odobranie) príjemcu skupinového vysielania.

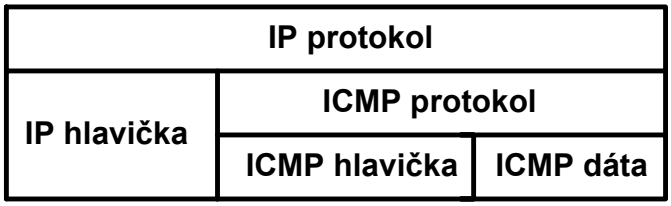
- **OSPF** (Open Shortest Path First) – OSPF protokol je smerovací protokol, ktorý používa smerovače k vytvoreniu smerovacích tabuliek. A na základe údajov z tabuliek sa následne rozhoduje, kadiaľ sa majú posielať pakety (IP protokolu) k požadovanému cieľu [4]. Ďalšie smerovacie prtokoly sú napriklad: RIP (Routing Information Protocol), IGRP (Interior Gateway Routing Protocol), EIGRP (Enhanced Interior Gateway Routing Protocol), apod.

- **IPsec** (IP security) – Táto sada protokolov slúži k zabezpečeniu komunikácie IP protokolom.

2.2.1 Protokol ICMP (Internet Control Message Protocol)

Je jeden z najdôležitejších protokolov zo sady protokolov Internetu. Používajú ho hlavne operačné systémy počítačov v sieti pre odosielanie chybových správ, napr. na oznámenie, že požadovaná služba neni dostupná alebo že smerovač alebo potrebný PC neni dosiahnuteľný.

Protokol ICMP je služobný protokol, ktorý je súčasťou IP protokolu. Protokol ICMP slúži k signalizácií mimoriadnych udalostí v sieťach postavených na IP protokolu [6]. ICMP paket býva zabalený do protokolu IP, teda do jeho dátovej oblasti. Na obr. 7 je zobrazený IP paket, ktorý obsahuje ICMP dátový paket.



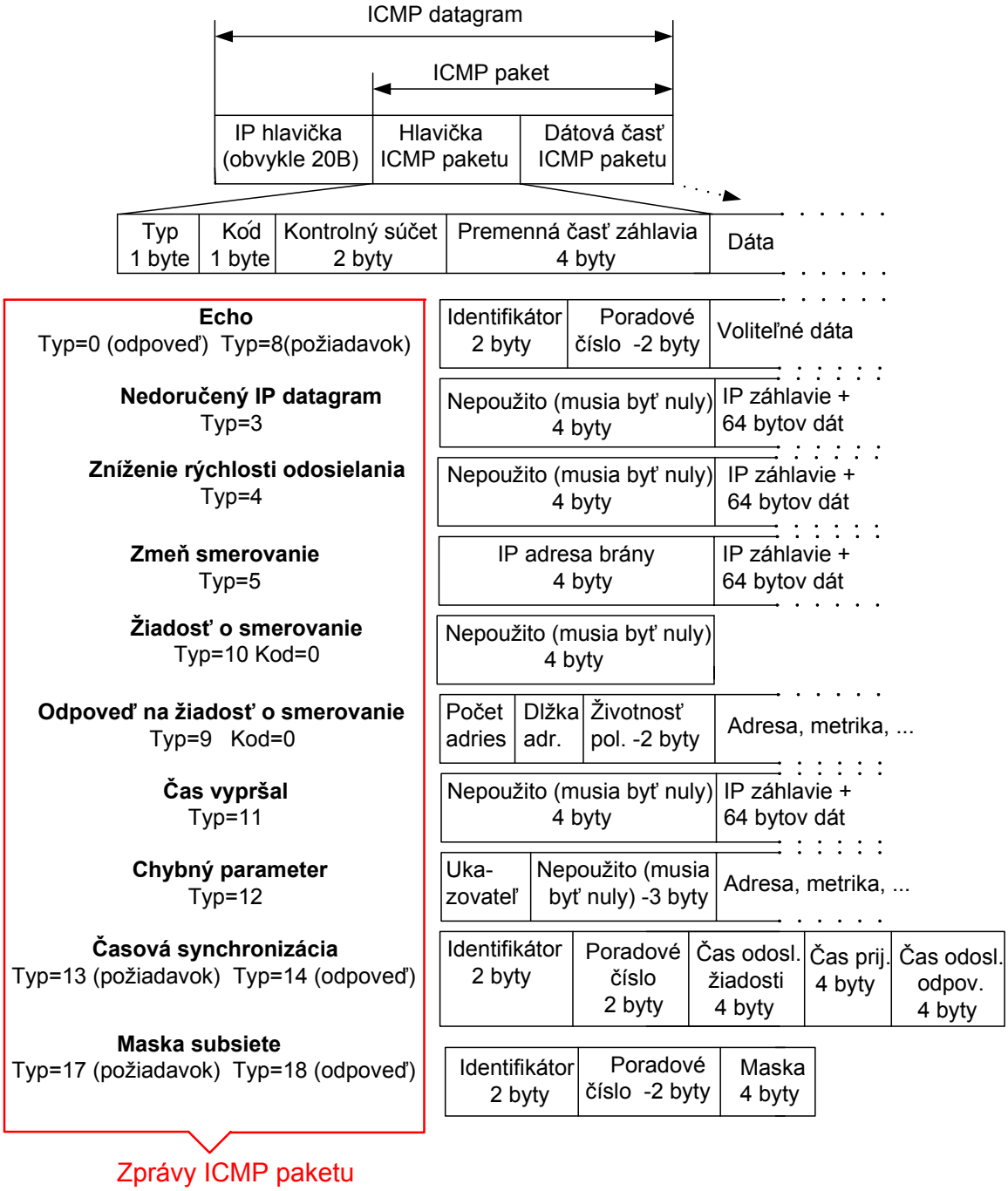
Obr. 7: Celková štruktúra ICMP paketu

Protokol ICMP môžeme signalizovať najrôznejšie situácie, ale v praxi bývajú mnohé z nich z bezpečnostných dôvodov zahadzované. Na obr. 8 je znázornená štruktúra hlavičky paketu. Hlavička ICMP paketu býva vždy dlhá 8 bytov.

Obr. 8: Štruktúra ICMP hlavičky

- **Typ** – Udáva jednoznačný typ ICMP paketu.
- **Kód** – Detailnejšie specifikuje typ ICMP paketu.
- **Kontrolný súčet** – Zaisťuje ochranu paketu pred poškodeným.
- **Premenná časť záhlavia** – V týchto štyroch bytoch sú umístnené ďalšie atirbúty. Ich názvy a hlavne veľkosti sú závislé na typu paketu.

Na obr. 9 je znázornená štruktúra hlavičky ICMP paketu a zprávy ICMP paketu.



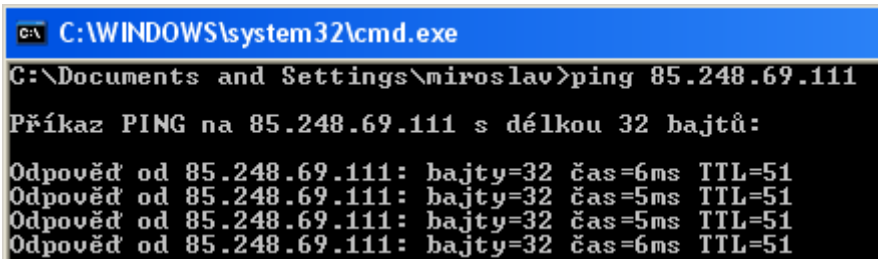
Obr. 9: Štruktúra a zprávy ICMP paketu [6]

Význam „identifikátora“ v hlavičke ICMP paketu spočíva v spárovaní požiadavok a odpovedí (teda povedané, aby sme mohli zistiť ku ktorému požiadavku patrí príslušná odpoveď).

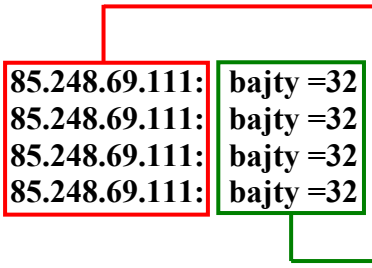
2.2.2 Zprávy ICMP paketu

2.2.3 Echo

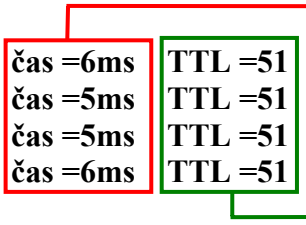
Je jednoduchý nástroj protokolu ICMP, pomocou ktorého môžeme testovať dosiahnu teľnosť uzlov v Internete [6]. Všetky operačné systémy podporujúce protokol TCP/IP obsahujú program ping, ktorý posiela ICMP zprávy typu 8 (Echo request) na cieľové zariadenie. Ak toto cieľové zariadenie túto správu obdrží, tak odpovie zdrojovému zariadeniu ICMP zprávou typu 0 (Echo reply). Ak zdrojové zariadenie túto správu obdrží, tak cieľové zariadenie je dostupné. Na obr. 10 je pomocou nástroja ping zistená dostupnosť uzlu 85.248.69.111 (www.sme.sk).



Obr. 10: Zistenie dostupnosti uzlu 85.248.69.111



Od stanice s IP adresou 85.248.69.111 prišla vždy ICMP ECHO odpoveď na každý ICMP ECHO požiadavok. Celkovo boli odoslané štyri ICMP ECHO požiadavky. V každom požiadavku bolo celkom 32 bajtov náhodných dát.

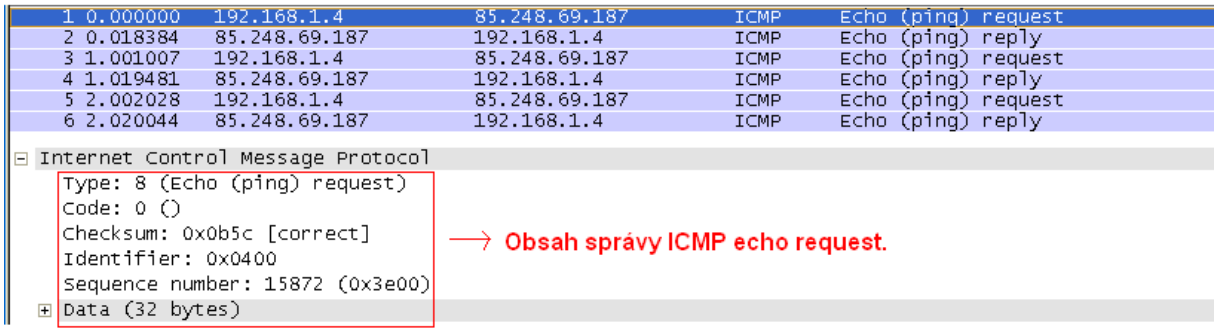


RTT – je čas od vysielania ICMP ECHO request do doby prijímu ICMP ECHO reply. V odpovedi mala položka TTL hodnotu 51.

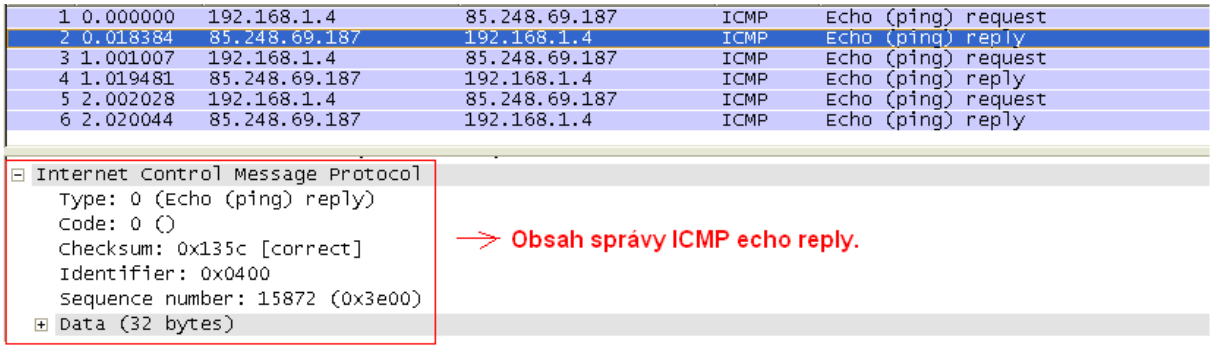
Definícia TTL:

- **Doba života datagramu (TTL – Time To Live)** – *Dobu života určuje odosielateľ. Každý smerovač pri zpracovávaný paketu zníži túto hodnotu o 1. V prípade „zatúlania“ paketu v sieti ho TTL-tý smerovač zničí* [7].

Na obr. 11 a 12 sú zachytené ICMP správy typu „ICMP echo request“ a „ICMP echo reply“ v programe Wireshark. Wireshark slúži k analýze komunikácie na sieťovom rozhraní. Má veľké množstvo filtrov pre obmedzenie zobrazenia rôznych sieťových protokolov.



Obr. 11: Obsah správy typu ICMP echo request zachytenej vo Wireshraku



Obr. 12: Obsah správy typu ICMP Echo reply zachytenej vo Wireshraku

Na obr. 13 je znázornený obsah správy ICMP echo request.

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Typ=8							**Kód=0**								**Kontrolný súčet**															
Identifikátor															**Poradové číslo**															
Dáta ….																														

Obr. 13: Obsah správy ICMP Echo request

Na obr. 14 je znázornený obsah zprávy ICMP echo reply.

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Typ=0							**Kód=0**								**Kontrolný súčet**															
Identifikátor															**Poradové číslo**															
Dáta ….																														

Obr. 14: Obsah správy ICMP Echo reply

2.2.4 Nedoručený IP datagram (Destination unreachable)

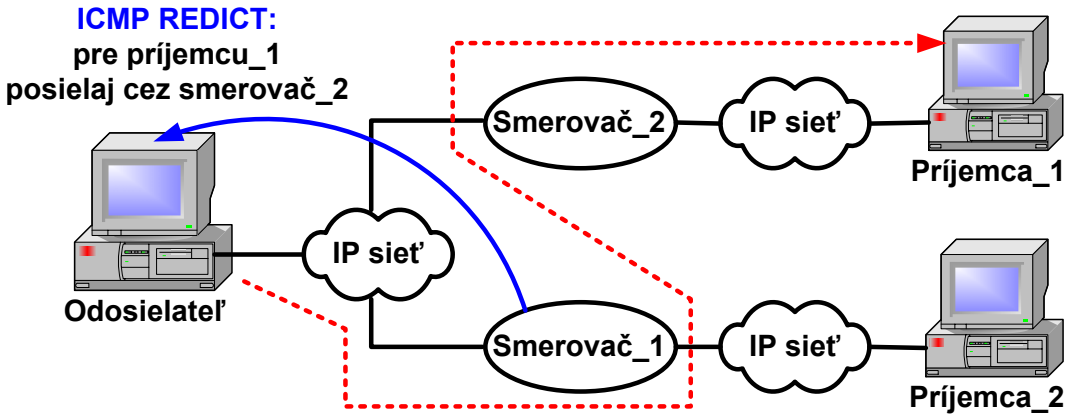
Ak IP datagram nemôže byť posielaný k adresátovi, tak je zahodený a odosielateľ je následne ICMP protokolom o tom oboznámený správou „Nedoručený IP datagram“.

2.2.5 Zníž rýchlosť odosielania (Source quench)

Ak je sieť medzi odosielateľom a príjemcom na niektorom mieste preťažená, tak smerovač, ktorý nie je schopný ďalej posielať IP datagramy signalizuje odosielateľovi „Zníž rýchlosť odosielania“ [6]. Ak odosielateľ používa protokol TCP (Transmission Control Protocol), tak zníži rýchlosť odosielania TCP segmentov a v prípade protokolu UDP (User Datagram Protocol) sa správy „Zníž rýchlosť odosielania“ ignorujú.

2.2.6 Zmeň smerovanie (Redirect)

Pomocou protokolu ICMP sa prevádzajú dynamické zmeny vo smerovacej tabuľke. Mechanizmus „Zmeň smerovanie (Redirect)“ je znázornený na obr. 15.



Obr. 15: Zmeň smerovanie [8]

Smerovač_1 sa stará o správne doručenie IP datagramu k príjemcovi_1 a to tak, že najskôr posiela dáta na smerovač_2 a ten sa postará o ich doručenie. Smerovač_1 posiela odosielateľovi správu „ICMP Redirect“, v ktorej žiada aby IP datagrami pre príjemcu_1 boli posielané cez smerovač_2. Odosielateľ by sa mal z toho poučiť a mal by si zapísať do svojej smerovacej tabuľky smerovač_2 a neskôr ho použiť.

2.2.7 Žiadosť o smerovanie

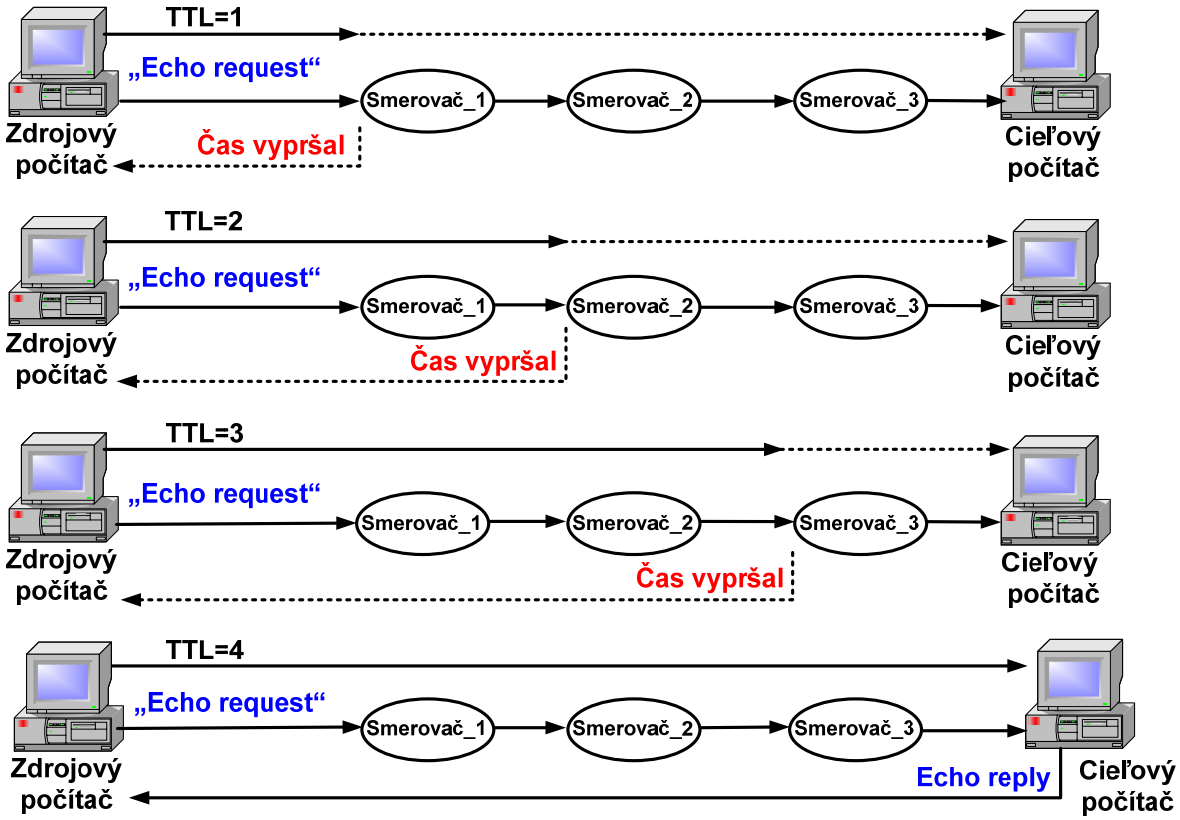
V podstate sa jedná o to, že nemusíme do smerovacej tabuľky počítačov na LAN ručne konfigurovať žiadnu položku default. Počítač ihneď po svojom štarte posiela obežníkom „Požiadavok o smerovanie“ a smerovač mu odpovie ICMP paketom „Odpoveď na žiadosť o smerovanie“, ktorá obsahuje: dĺžku adresy, počet adries smerovača a potom dvojicu IP adresa a preferencie. A z odpovedí môže počítač vygenerovať automaticky položku default. Ak má preferencia vyššiu hodnotu, tak je IP adresa viac preferovaná. Ak je hodnota preferencie 80000000_{16}, tak to signalizuje, že táto adresa sa má zo smerovacej tabuľky vypustiť. Smerovače odpovedajú na žiadosť o smerovanie, avšak v náhodnom intervale medzi 450 a 600 sekundami by mali obežníkom sami do lokálnej siete generovať ICMP pakety „Odpoveď na žiadosť o smerovanie“. Položka „doba života“ udáva čas, po ktorý je informácia platná [6].

2.2.8 Čas vypršal (Time exceeded)

Tento typ zahrňuje dva odlišné prípady:

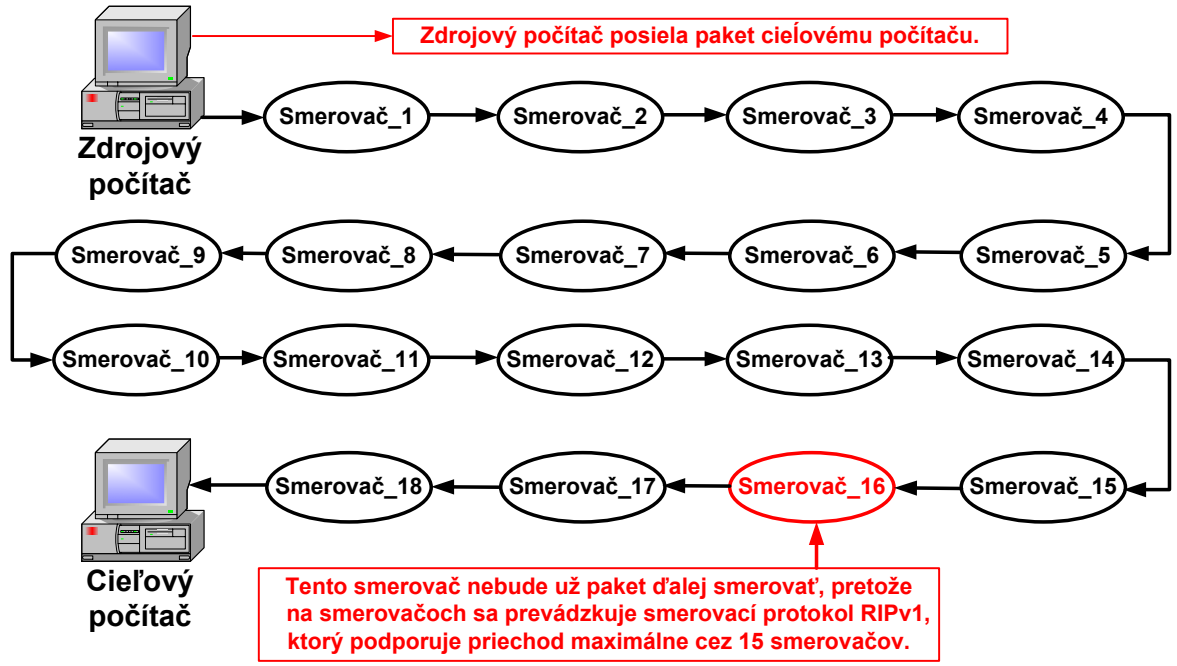
1. Pre kód=1 signalizuje, že hodnota TTL bola na smerovači znížená na nulu, tak bude paket zahodený [6].
2. Pre kód=1 signalizuje, že počítač adresáta nedokáže v danom časovom intervale z fragmentov zostaviť celý IP datagram [6].

ICMP paket „Čas vypršal“ využíva ku svojej činnosti program tracert. Program tracert odosiela zo zdrojového počítač na cieľový uzol ICMP pakety „Echo request“, ale v prvom pakete nastaví položku TTL na jednotku. Hneď prvý smerovač na ceste zahodí paket a vráti ICMP paket „Čas vypršal“, preto lebo musí TTL zmenšiť o jednotku, ale týmto zmenšením už dostane nulu. Zdrojový počítač tak od prvého smerovača na ceste dostane v IP datagrame ICMP paket „Čas vypršal“. Z položky adresa odosielateľa v IP záhlaví ide zistiť adresu prvého smerovača na ceste. Zmeria sa tak časový interval od dosielania paketu po príjem paketu a zistí sa tak čas paketu od odielateľa ku príjemcovi a naspäť. Tento proces sa opakuje trikrát a všetky tieto tri časy sa zobrazia. Na konci riadku ešte zobrazuje meno a IP adresu smerovača . Meno získa z reverzného prekladu v DNS. Fungovanie DNS systému je popísané v kap. 6.1. Ak nezíska v časovom limite odpoveď, tak sa zobrazí namiesto času hviezdička (*). Tento postup sa opakuje ďalej akurát bude hodnota TTL =2 atd. Svoju činnosť ukonči v okamžiku, keď od cieľového uzlu obdrží ICMP správu “Echo reply”. K ukončenie môže dôjsť aj vtedy ak nejaký smerovač nepozná cestu k cieľovému počítači, tak zdrojovému počítaču posiela správu “ Nedoručený IP datagram”. Proces, ktorý tu je popísaný je znázornený na obr. 16.



Obr. 16: Príkaz tracert

Smerovací protokol RIP (Rounting Information Protocol) má smerovaciu medz (TTL), ktorou môže paket prejsť 15, což znamená, že paket bude schopný cestovat maximálne cez 15 smerovačov viz obr. 17. Na všetkých smerovačoch je prevádzkovaný protokol RIPv1 (Rounting Information Protocol verzie 1).



Obr. 17: Príklad ICMP paketu

Na obr. 18 je znázornený výpis trasy z uzlu 147.229.192.1 [b05-sm.kn.vutbr.cz] do uzlu 85.248.69.111 [www.sme.sk] pomocou programu tracert v prostredí Windows. Tracert je program, ktorý vypisuje uzly (res. smerovače) na ceste paketov od zdrojového počítača k cieľovému počítaču.

```
Příkazový řádek
C:\Documents and Settings\miroslav>tracert www.seznam.cz

Výpis trasy k www.seznam.cz [77.75.76.3]
s nejvýše 30 směrováními:

  1    < 1 ms    < 1 ms    < 1 ms  b05-sm.kn.vutbr.cz [147.229.192.1]
  2    < 1 ms    < 1 ms    < 1 ms  hp-list.net.vutbr.cz [147.229.252.118]
  3    < 1 ms    < 1 ms    < 1 ms  hp-list2.net.vutbr.cz [147.229.252.73]
  4    < 1 ms    < 1 ms    < 1 ms  hp-kou.net.vutbr.cz [147.229.252.33]
  5    < 1 ms    < 1 ms     1 ms  hp-ant.net.vutbr.cz [147.229.254.14]
  6    < 1 ms    < 1 ms    < 1 ms  fw-ant.net.vutbr.cz [147.229.254.230]
  7    < 1 ms    < 1 ms    < 1 ms  hp-ant2.net.vutbr.cz [147.229.254.229]
  8    < 1 ms    < 1 ms    < 1 ms  r98-bm.cesnet.cz [147.229.252.17]
  9     4 ms      4 ms      4 ms  nix2.seznam.cz [194.50.100.194]
 10     4 ms      4 ms      4 ms  www.seznam.cz [77.75.76.3]

Trasování bylo dokončeno.
```

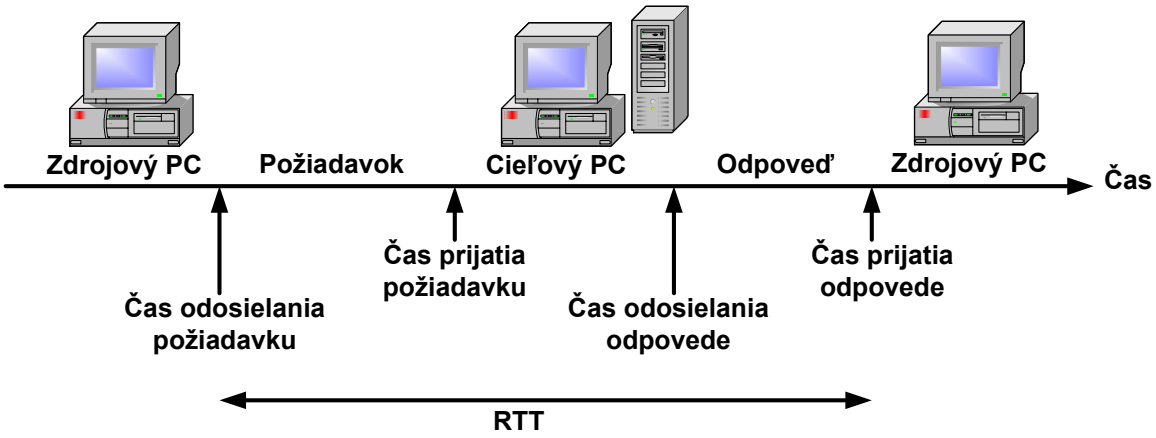
Obr. 18: Ukážka programu tracert v prostredí Windows

2.2.9 Požiadavka o masku (Address Mask Request)

Ako náhle ako stanica pomocou protokolu RARP (Reverse Address Resolution Protocol) obdržala svoju IP adresu, tak môže žiadať o masku svojej siete.

2.2.10 Časová synchronizácia (Time synchronization)

ICMP paketom žiada cieľový počítač o čas. Na obr. 19 je znázornený mechanizmus časovej synchronizácie. Zdrojový počítač do ICMP paketu „Požiadavok na časovú synchronizáciu (Timestamp request)“ vyplní čas odosielania požiadavkua a cieľový počítač vyplní do svojej odpovede „Odpoveď na časovú synchronizáciu (Timestamp reply)“ dva časy: 1, Čas prijatia požadavku a 2, Čas odosielania odpovede [6].



Obr. 19: Časová synchronizácia [6]

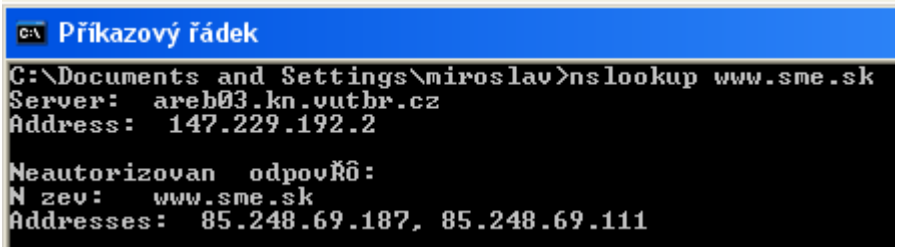
Zdrojový počítač si zistí čas prijatia odpovede (tento čas sa neprepravuje v žiadnom ICMP pakete). Doba RTT (Round Trip Time) sa získa odčítaním času odosielaného požiadavku od času prijatia odpovede. Doba RTT je doba od zdrojového počítača k cieľovému počítaču a späť.

3. Útoky na DNS a systémy smerovania paketov

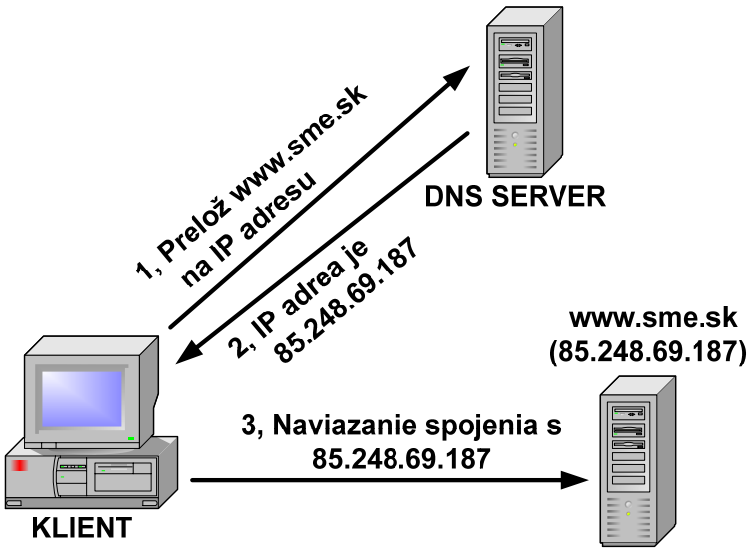
3.1 Systém DNS (Domain Name System)

DNS je systém na správu doménových mien počítačov a ich IP adries. Umožňuje preklad doménového mena na IP adresu [6]. Jedná sa o to, že užívatelia si IP adresy veľmi ťažko pamätajú, tak preto sa využíva namiesto IP adresy názov sieťového rozhrania. Preto pre každú IP adresu máme zavedené meno sieťového rozhrania (počítača), inak povedané doménové meno. Na zistenie sieťového rozhrania z IP adresy môžeme použiť napríklad príkaz: nslookup www.sme.sk. viz obr. 20.

Jednej IP adrese môžeme priradiť aj niekoľko doménových mien. V DNS databáze je definovaná väzba medzi menom počítača a IP adresou. DNS je celosvetová distribuovaná databáza. Jednotlivé časti tejto databáze sú umiestené na tzv. name serveroch (na našom obr. DNS serveroch). Fungovanie DNS systému je zobrazené na obr. 21. DNS je založený na modelu klient – server a pracuje na aplikačnej vrstve. Používa porty TCP/53 i UDP/53.



Obr. 20: Zistenie IP adresy zo sieťového rozhrania www.sme.sk



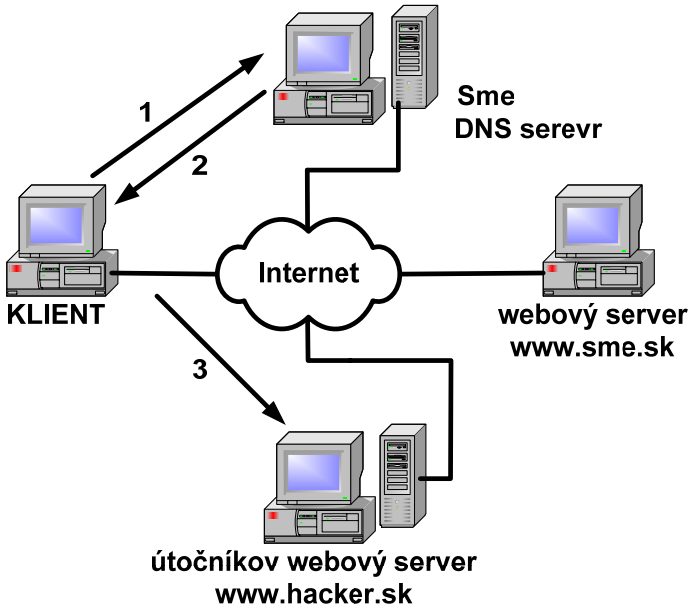
Obr. 21: Spôsob komunikácie v DNS systéme [4]

3.2 Útoky ohrozujúce smerovanie paketov

Tieto útoky sú založene na manipulácií so smerovacími tabuľkami a spôsobujú zneprístupnenie služieb sieťam [9]. Smerovacie protokoly (napr. RIPv1) majú väčšinou slabú alebo žiadnu autentizáciu. Pri nedostatočnej autentizácií umožňuje útočníkom zmeniť smerovacie tabuľky (pomocou potvrdenia autentizačného údaja, ktorým môže byť napríklad IP adresa) a presmerovať tak dátový tok do svojej siete.

3.3 Útoky na DNS servery

Útoky na DNS (Domain Name System) servery sú podobné ako útoky na smerovacie tabuľky. Väčšina týchto DNS útokov spočíva v umiestený nesprávnej informácie do cache nameserveru. Tento nameserver následne poskytuje informácie, ktoré môžu klienty nasmerovať na iný server (ako oficiálny) [10]. Na obr. 22 je znázornený mechanizmus útoku, krorý spôsobuje dlhotrvajúcu nedostupnosť serverov.



Obr. 22: Infiltrácia do DNS cache [10]

1. Klient sa chce pripojiť na webový server Sme. Resolver (resolver – je komponent systému, ktorý spolupracuje s DNS) webového prehliadača požiada DNS server o IP adresu www.sme.sk.

2. Cache DNS serveru je nainfikovaný útočníkom, takže klient dostane namiesto IP adresy serveru Sme, IP adresu serveru www.hacker.sk.

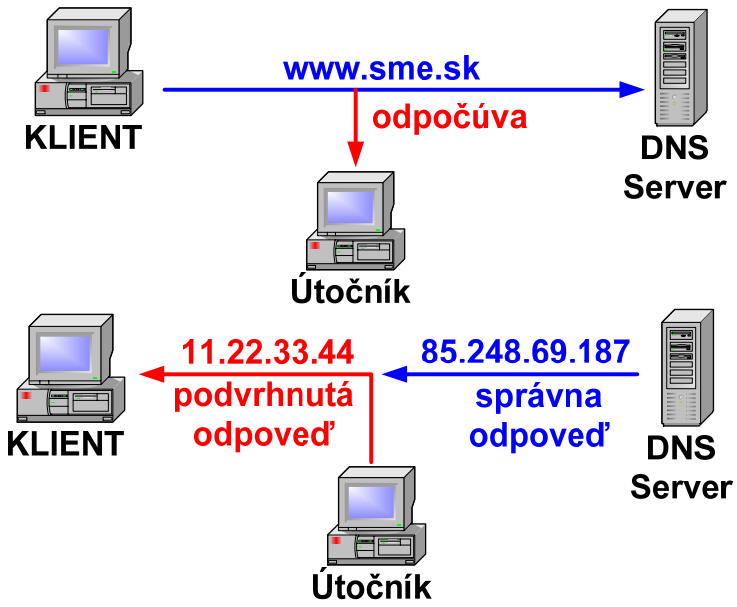
3. Systém útočníka je teda klientom považovaný za server.

3.4 Útok typu DNS spoofing (falšovanie DNS údajov)

Tento typ útoku sa odhaľuje veľmi zle, hlavne vtedy ak sa útočník nachádza medzi klientom a DNS serverom [11]. DNS sa dá potom zabezpečiť pomocou DNSSEC (DNS Security Ex-

tensions). Jedná sa o rozšírenie DNS o asymetrickú kryptografiu a digitálne podpisy DNS zón. Ak klient s týmto rozšírením dostane odpoveď od DNS serveru, môže si podľa digitálneho podpisu zóny overiť autencititu a integritu odpovedí. Ak nastane situácia, že namiesto podpísaných odpovedí dostane odpoveď nepodpísanú, jednoducho ju zahodí.

Na obr. 23 je znázornený útok typu DNS spoofing.



Obr. 23: Útok typu DNS spoofing

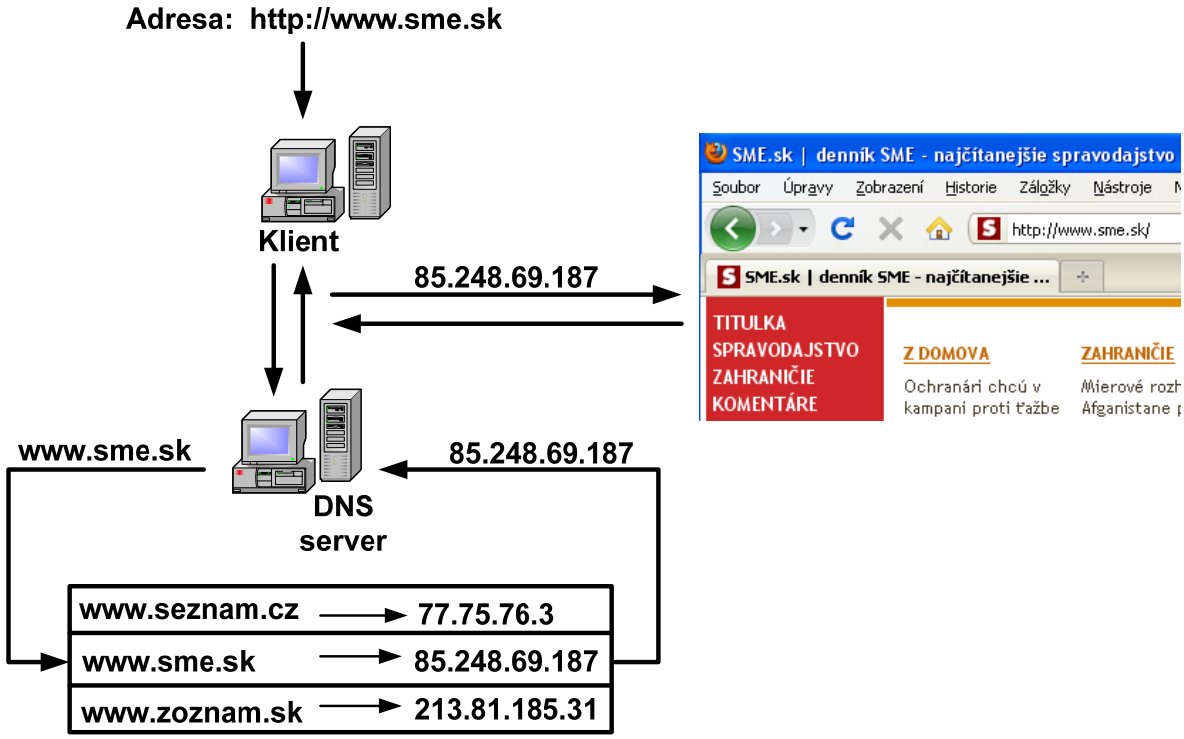
Klient chce zobraziť webové stránky www.sme.sk. Do internetového prehliadača napíše adresu. Priehliadač najskôr kontaktuje DNS server, aby mu zistil IP adresu odpovedajúcu www.sme.sk. Následne sa prehliadač pripojí k serveru s danou IP adresou a stiahne webovú stránku.

Útok je prevedený tak, že útočník počúva DNS požiadavky a ak nejaký zachytí, odpovie najskôr falošnou IP adresou ako požadovaný DNS server. Na obr. 21 klient vysiela požiadavok k DNS serveru za účelom zistiť IP adresu z názvu www.sme.sk, ale útočníkovi sa podarí odpočuť požiadavok a posiela odpoveď s inou IP adresou. Odpoveď od DNS serveru, ktorá príde neskôr klient ignoruje.

3.5 Útok typu DNS poisoning (otrava DNS)

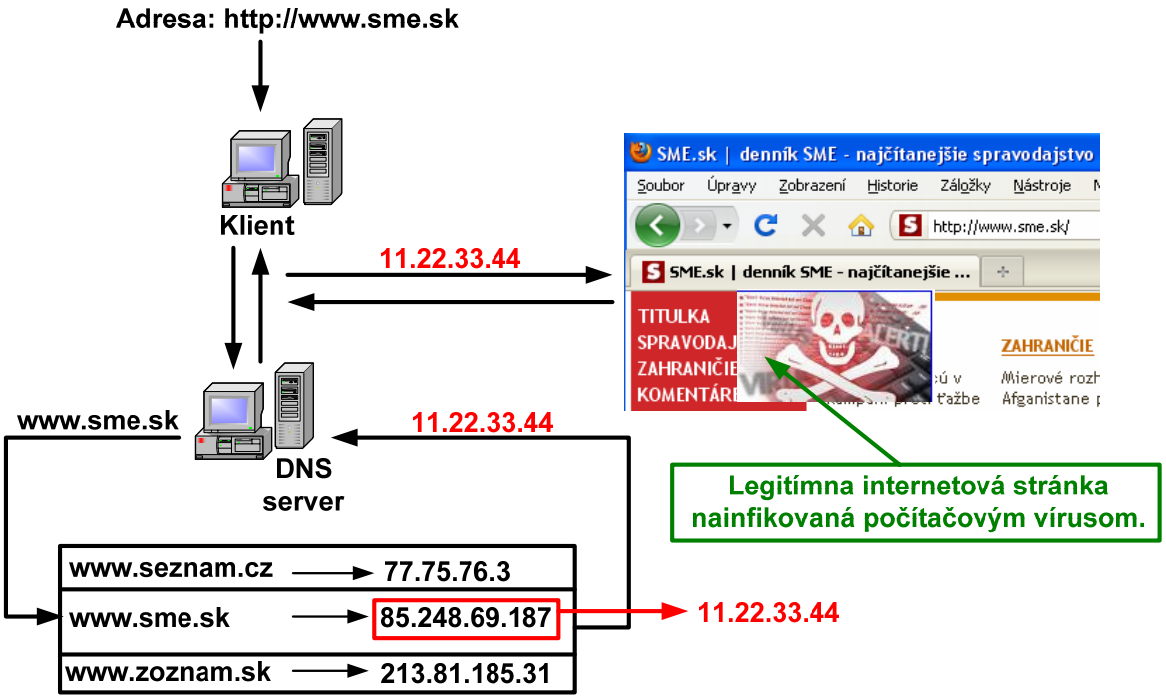
Tento typ útoku dokáže oklamať DNS klienta na počítači tým, že mu podsúva falošné informácie, ktoré DNS klient považuje za pravdivé. Tieto nepravdivé informácie si DNS klient uchováva na určité časové obdobie uchováva v pamäti cache [12]. Napríklad útočník môže zmanipulovať DNS záznamov IP adries a vytvoriť tak u uživateľa dojem, že navštevuje legitímnu internetovú stránku, no ale v skutočnosti mu môže byť podsunutý napr. počítačový vírus viz obr. 25.

Na obr. 24 je znázornená komunikácia medzi klientom a DNS serverom.



Obr. 24: Komunikácia medzi klientom a DNS serverom

Na obr. 25 je znázornený útok typu DNS poisoning.



Obr. 25: Útok typu DNS poisoning

4. Návrh a praktická realizácia robota

Pri návrhu a realizácií robota bol použitý programovací jazyk Jáva s vývojovým prostredím NetBeans 6.8.

4.1 Programovací jazyk Java a vývojové prostredie Netbeans IDE 6.8

Jáva je objektove orientovaný jazyk, ktorý vyvíja spoločnosť Sun Microsystems, Inc. Java bola navrhnutá tak, aby bola ľahko prenositeľná na rôzne počítačové platformy. Na rozdiel od *C++* sa v prípade Javy zdrojový text *kompiluje* do strojovo nezávislého, veľmi efektívneho bajtového kódu. Ten sa potom interpretuje prostredníctvom modulu JVM (Java Virtual Machine) na ľubovoľnej počítačovej platforme, podporujúcej tzv. Java-runtime, (*Windows 95/NT 4.0*, *Sun OS 4.1*, *Sun Solaris 2.4*, a.i.). Javovské programy tak môžu byť prostredníctvom Internetu presúvané z jedného počítačového systému na druhý bez akejkoľvek transformácie a bez akéhokoľvek zásahu užívateľa. Pri návrhu jazyka bol kladený dôraz na bezpečnosť. Preto ešte pred spustením bajtového kódu je preverená jeho syntax, čím je vylúčené spadnutie programu spôsobené poškodeným kódom. Javovský program nemá prístup ani k lokálnym programom, ani k lokálnym zdrojom, čo podľa jeho tvorcov vylučuje nebezpečenstvo javovských vírusov“.

Prostredie NetBeans 6.8 je určené pre vývoj aplikácií v jazyku Jáva, ale podporuje i dalšie programovacie jazyky (napr. C++, PHP). NetBeans IDE je bezplatne šířený produkt, ktorý ide používať bez akýhkoľvek obmedzení. Okrem vývojového prostredia je tiež dostupná vývojová platforma NetBeans Platform, čo je rozšiřiteľný základ pre použitie pri vytváraní rozsiahlých aplikácií. Program ide rozšířiť pomocou doplňujúcich modulov [13, 14, 15].

4.3 Návrh a realizácia robota pracujúceho s protokolom HTTP

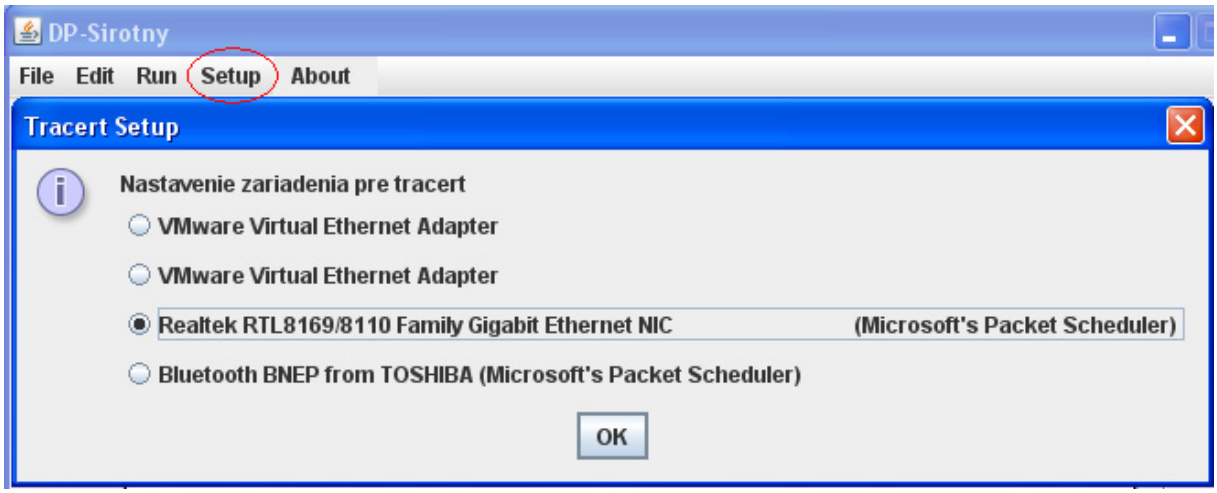
Vytvorený robot je určený k prieskumu a mapovaniu topológie počítačovej siete ľubovolnej veľkosti na aplikačnej vrstve a to na protokole HTTP. Na obr. 26 je znázornené vytvorené menu programu pre našu aplikáciu.

Obr. 26: Vytvorené menu programu

- **Položka** (File) – Slúži na ukončenie celej aplikácie.
- **Položka** (Edit) – Slúži na uloženie odkazov do textového súboru.
- **Položka** (Run) – Slúži na spustenie aplikácie.
  – Slúži na ukončenie aplikácie.
- **Položka** (Setup) – Slúži na vybratie sieťoveho rozhranie.
- **Položka** (About) – Je to informácia o vytvorenej aplikácie.

Metóda SetupTracert slúži na vybratie sieťového rozhrania viz obr. 27.

```
public static void setupTracert(final MainWindow parent) {
   jpcap.NetworkInterface[] devices = JpcapCaptor.getDeviceList();
      .....
}
```



Obr. 27: Metóda SetupTracert slúži na vybratie sieťového rozhrania

Na obr. 28 je znázornené fungovanie metódy „Run“. Je to akcia na stlačenie tlačítka „START“ a táto akcia spúšťa thread, v ktorom beží samotné prehľadávanie a do textového pola „jTextField“ sa načítava testovaná stránka http://www.sme.sk.

```
public void run()  {
......
}
```

Obr. 28: Fungovanie metódy Run

Hranatá zátvorka na obr. 28 pred výpisom URL (Uniform Resource Locator) označuje hĺbku vnorenia. Podľa toho sa dá vyhľadať, ktorá stránka je tzv. materská. V našom prípade je materská stránka http://www.sme.sk. V ďalšom texte si vysvetlíme fungovanie metódy „Read“.

Metóda „Read“ zabezpečuje vytvorenie BufferedReader (tzv. je to návratový typ metódy) objektu zo zadanej URL adresy. Najprv sa vytvorí objekt URL zo zadanej adresy, vytvorí sa konekcia a z nej získame InputStream. Z toho ďalej cez InputStreamReader sa dostaneme až k objektu BufferedReader. BufferedReader umožňuje efektívne čítanie znakov, polí a riadkov nakoľko využívame zásobník (buffer) – je to vyrovnávacia pamäť, ktorá je určená pre dočas-

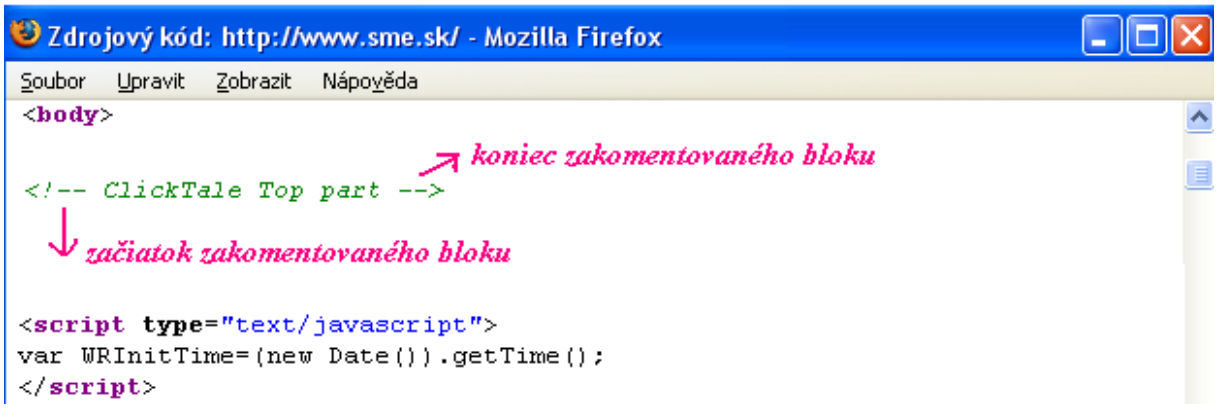
né uchovanie dát pred ich presunutím na iné miesto. To využijeme v metóde FindLinksOnPage, v ktorej je metóda read volaná a vracia BufferedReader.

```
public static void FindLinksOnPage(String link, int level,
      String domainAdress, LinkDataHandlerClass mnozinaPrehladanychLinkov,
      LinkDataHandlerClass mnozinaNaPrehladanie) throws Exception {
        BufferedReader reader = null;
           try {
              reader = ParseFunctions.read(link);

              String line = reader.readLine();
```

K prehliadaniu jednotlivých riadkov na stránke slúži metóda „Parse“. Metóda slúži k získaniu informácií z reťazca. Ak chceme čítať dáta napríklad zo vstupu, sú všetky reprezentované ako string. Ak sú zas dáta číselné a ak potrebujeme získať ich hodnotu, tak ju získame pársovaním vstupného reťazca. V našej aplikácií metóda slúži na spracovanie načítaného riadku webovej stránky a to tak, že zisťujeme či sa nachádzame v zakomentovanom bloku alebo sa nenachádzame v zakomentovanom bloku viz obr. 29.

```
public static void Parse(String line, int level, String base_adress,
     String domainAdress,
     LinkDataHandlerClass mnozinaPrehladanychLinkov,
     LinkDataHandlerClass mnozinaNaPrehladanie) {
.....
}
```



Obr. 29: Ukážka zakomentovaného bloku

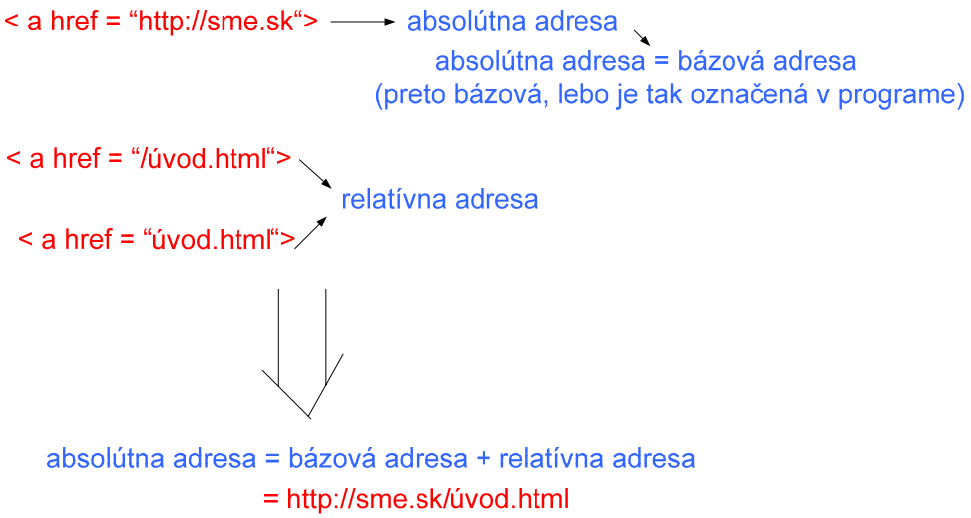
Ak sa nachádzame v za komentovanom bloku, tak sa snažíme nájsť koniec zakomentovaného bloku viz obr. 26. Ak nenájdeme ukončovaciu značku komentára, tak metódu ukončíme a ak nájdeme ukončovaciu značku komentára, tak sa zruší príznak, že sme v komentovanom bloku a parsuje zvyšok nezakomentovaného riadku. Ďalej je tu inicializovaná metóda void SetLinkName (), ktorá slúži na vybratie odkazu z riadku. Pomocou tejto funkcie sa v riadku vyhľadajú prve a druhé úvodzovky. Ďalej ak sa v odkaze nenachádza odkaz na stránku, teda povedané „javascript“ ale respektívne sa tam nachádza odkaz na mailovú adresu, tak ukončí metódu. Ďalej pri prehladávaní stránky sa tam môžu vyskytnúť tzv. Sessions stranky, sú to stránky, ktoré su automaticky vygenerované pre konkrétnu situáciu (tú totiž môže ísť o rovnakú stránku, ale mohlo by dôjsť k zacykleniu pri novom vytváraní session stránok zo strany servera pri každom ďalšom volaní, pričom môže ísť o stránku s rovnakým odkazom). Tieto session

stránky sa hľadajú podľa „**otáznika**“ (**?**). Ak je otáznik v linku, tak link skrátime iba po otáznik napr. http://twitter.com/?status=http%3A%2F%2Fwww.cas.sk%2Fclanok%2F142972#search?q=Xmas a link skrátime teda na http://twitter.com/. Ďalej ak sa v odkaze nachádza na prvom mieste „**bodka**“ (**.**), tak tiež ukončí volanie metódy, pretože v tomto prípade nejde o odkaz na stranku (napr. **http://.www.vutbr.cz/**). Ďalej kontroluje či sa v linku nachádza „**lomená čiara**“ (/), ak sa nachádza a je na prvom mieste tak sa jedná o relatívnu URL (napr. **/úvod.html**). Program si samozrejme pamätá adresu stránky, ktorú práve spracováva (napr. nazvime ju bázová adresa) a keď načíta iba relatívnu adresu, tzv. keď sa na prvej pozícií v texte nachádza znak lomítka (/), alebo sa v texte nenachádza zkratka „http:“, vieme, že ide o relatívnu adresu a absolútnu adresu získame sčítaním bázovej adresy s relatívnou adresou. Rozdiel medzi relatívnou a absolútnou adresou je znázornený na obr. 27.

Ďalej je použitá funkcia „SetLinkName“, ktorá slúži na vybratie odkazu z riadku.

```
void SetLinkName(String line, int level, int position) throws Exception {
     .....
}
```

Na obr. 27 je znázornený rozdiel medzi relatívnou a absolútnom adresou.



Obr. 27: Rozdiel medzi absolútnou a relatívnou adresou

V predchádzajúcom texte sme si vysvetlite fungovanie metód Read, Parse a SetLinkName. Pomocou týchto metód dokážeme na stránke „odkaz (link)”: vyhľadať, spracovať a vybrať. Na základe znalostí týchto metód si vysvetlíme fungovanie metódy „FindLinksOnPage“.

```
public static void FindLinksOnPage(String link, int level,
 String domainAdress, LinkDataHandlerClass mnozinaPrehladanychLinkov,
 LinkDataHandlerClass mnozinaNaPrehladanie) throws Exception {

          BufferedReader reader = null;
               try {
                   reader = ParseFunctions.read(link);

                   String line = reader.readLine();
```

Legenda:

- **MnožinaNaPrehladanie** – Obsahuje linky ktoré sa majú ešte prehľadať.
- **MnožinaPrehladanýchLinkov** – Obsahuje linky, ktoré sme už prehľadali.

Deklarované sú nasledovne:

LinkDataHandlerClass mnozinaNaPrehladanie ;
LinkDataHandlerClass mnozinaPrehladanychLinkov ;

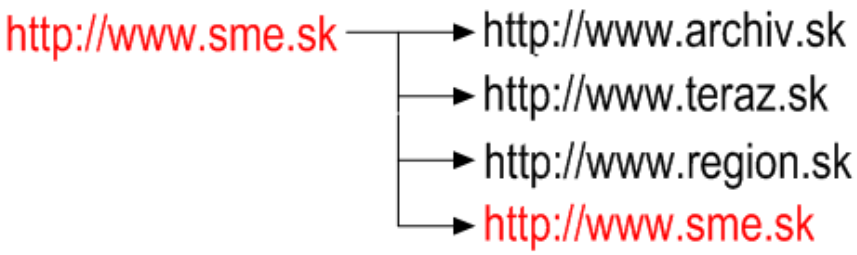
kde LinkDataHandlerClass je deklarovaná nasledovne:

class LinkDataHandlerClass extends TreeSet<String>

Prečo takýto zápis? Je možné ľahko zameniť kolekciu TreeSet sa inú napríklad HashSet alebo Stack, atď, ak by bolo potrebné experimentovať, alebo skúšať iné kolekcie. Kde „String“ hovorí o tom, že sa bude jedna o reťazce, a TreeSet je množina ktorá ma v sebe usporiadane prvky a garantuje log(n) časovú náročnosť pre základné operácie ako „add, remove, contains“.

Proces funguje tak, že keď nejakú stránku prehľadávame, tak ten link dáme do množiny PrehladanychLinkov a z množiny NaPrehladanie ho odstránime. A vždy keď sa pridáva link do množiny NaPrehladanie, tak pozrieme najprv do množiny PrehladanychLinkov, či tam už nie je, tzv. už bol prehľadaný, potom pozrieme do množiny NaPrehladanie, tzv. či už sme ho tam nezapísali v predchádzajúcom kroku, a ak nie, tak ho pridáme.

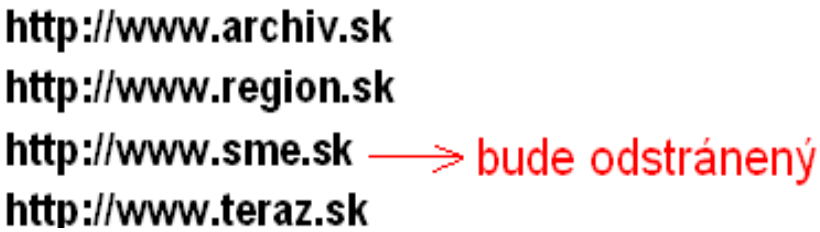
Na obr. 30 si uvedieme príklad fungovania metódy FindLinksOnPage.



Obr. 30: Príklad fungovania metódy FindLinksOnPage

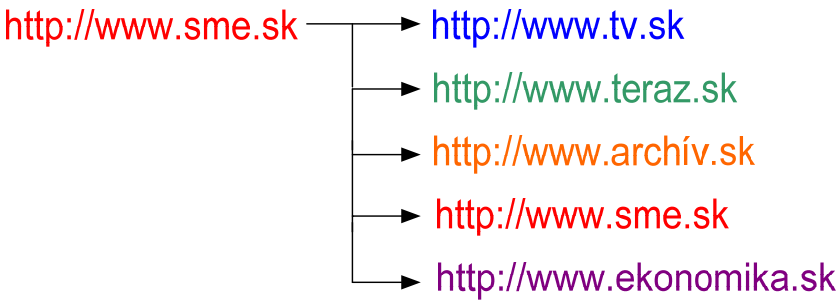
Proces spočíva v tom, že ak začíname na stránke http://www.sme.sk, tak sa z neho vypíšu všetky linky a uložia do množiny NaPrehladanie (tzv. vždy v abecednom poradí) viz obr. 31. Do množiny NaPrehladanie sa vypísali linky: http://www.archiv.sk, http://www.teraz.sk, http://www.region.sk a link http://www.sme.sk sa už nachádza v množine PrehladanychLinkov viz obr. 32 a bude teda z množiny NaPrehladanie odstránený. Ďalej ak link napr. http://www.archiv.sk nemá žiadny odkaz, tak vyberáme z množiny NaPrehladanie ďalší odkaz (vždy v abecedním poradí) v našom prípade je to htttp://www.region.sk.a začneme ten prehľadávať.

Obr. 32: Množina PrehladanychLinkov



Obr. 31: Množina NaPrehladanie

Na obr. 33 je znázornená ochrana proti zacykleniu.

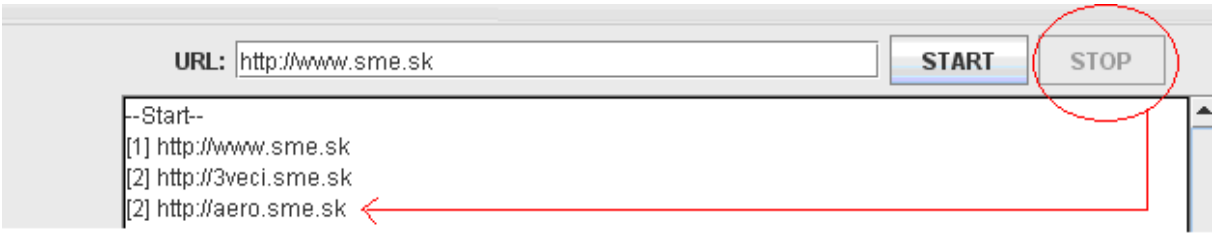


Obr. 33: Ochrana proti zacykleniu

Ochrana spočívá v tom, že ak začíname na stránke http://www.sme.sk, tak sa z nej postupne vypíšu všetky podstránky, ktoré sa tam nachádzajú: http://www.tv.sk, http://www.teraz.sk, http://www.archív.sk a http://www.sme.sk, ale ak táto stránka sa už v zázname nachádza, tak ju preskočíme a ideme na ďalšiu a to na stránku http://www.ekonomika.sk, keby sme ju nepreskočili tak by došlo k zacykleniu.

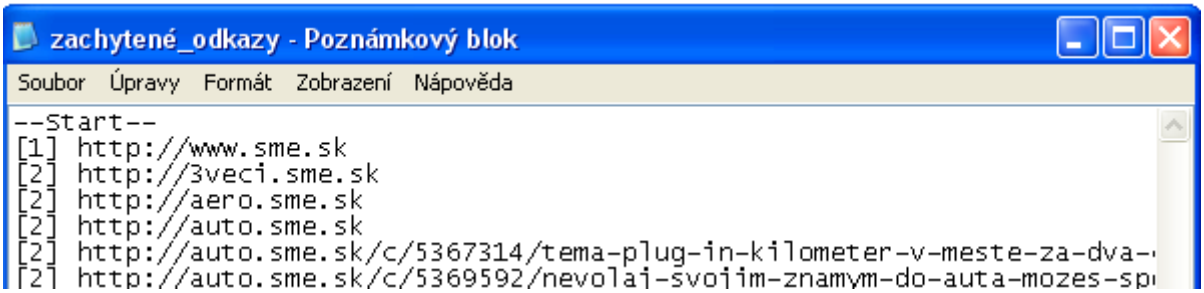
Fungovanie metódy „Stop“ je znázornené na obr. 34. Je to akcia na stlačenie tlačítka „STOP“, ktorá ukončuje thread, v ktorom beží samotné prehľadávanie a to tak, že sa prestanu vypisovať nové odkazy (linky).

```
public void stop()  {
......
}
```



Obr. 34: Fungovanie metódy Stop

Na obr. 35 je znázornené ako sa zachytené odkazy ukladajú do súboru zachytené_odkazy.txt.



Obr. 35: Ukladanie zachytených odkazov do súboru zachytené_odkazy.txt

4.4 Návrch a realizácia robota pracujúceho s protokolmi HTTP a ICMP

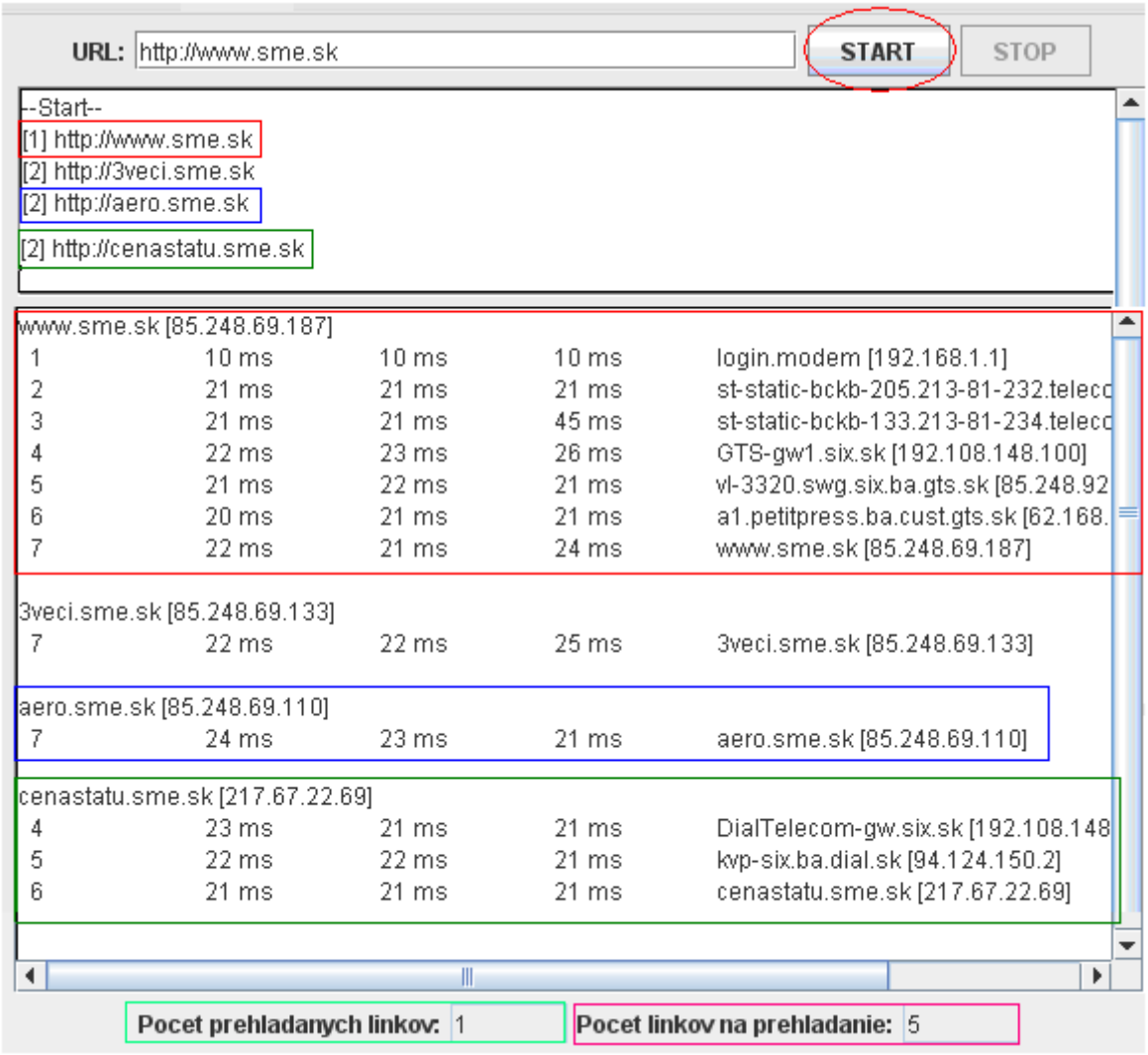
Vytvorený robot je určený k prieskumu a mapovaniu topológie počítačovej siete ľubovoľnej veľkosti a to na protokole HTTP a ICMP. Pri vytváraný tejto aplikácie je použitý program z kap. 5.3, ktorý slúži na zachytávanie odkazov z webovej stránky.

Na obr. 36 je znázornené vytvorené menu pre vytvorenú aplikáciu.

Obr. 36: Vytvorené menu programu

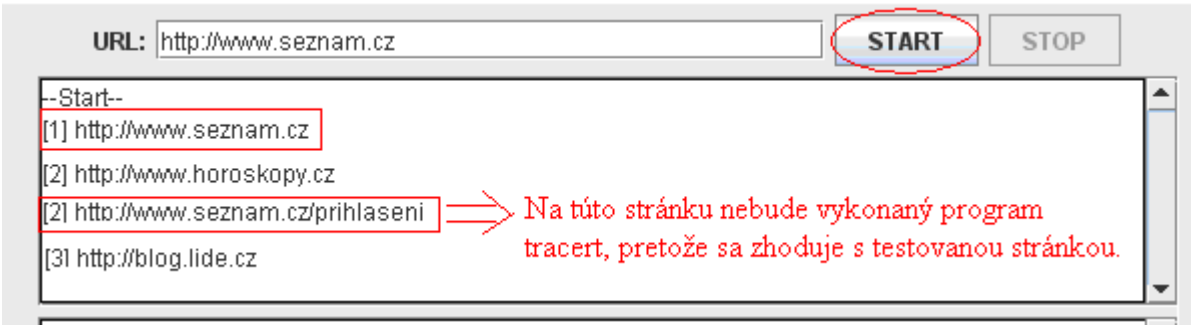
- **Položka** (File) – Slúži na ukončenie celej aplikácie.
- **Položka** (Edit) – Slúži na uloženie odkazov do textového súboru.
  – Slúži na ukončenie tracertov do textového súboru.
- **Položka** (Run) – Slúži na spustenie aplikácie.
  – Slúži na ukončenie aplikácie.
- **Položka** (Setup) – Slúži na vybratie sieťoveho rozhranie.
- **Položka** (About) – Je to informácia o vytvorenej aplikácie.

Na obr. 37 je vidieť ako náš vytvorený robot zachytáva odkazy z webovej stránky (vrchná časť programu) a následne sa na jednotlivé odkazy vykonáva program tracert (spodná časť programu). Na testovanú stránku www.sme.sk je vykonaný celý tracert [tzv. vypisuje cestu paketov od zdrojového počítača k cieľovému počítaču => tzv. cez aké uzly (smerovače) prechádza paket od zdroja k cieľu]. Na ďalšie zachytené odkazy je vykonaný tracert, ale už sa vypíšu len tie uzly (smerovače), ktoré sú odlišné oproti tracertu vykonaného na stránku www.sme.sk. Napríklad tracert vykonaný na stránku www.aero.sme.sk má prvých 6 uzlov rovnakých ako tracert vykonaný na stránku www.sme.sk a až od 7 uzlu sa líši => takže siedmi uzol bude vypísaný v našej aplikácií. Ďalej tracert vykonaný na stránku www.cenastatu.sme.sk má prvé 3 uzly rovnaké ako tracert vykonaný na stránku www.sme.sk a až od 4 uzlu sa líši => takže uzly 4,5,6 budú vypísané v našej aplikácií.



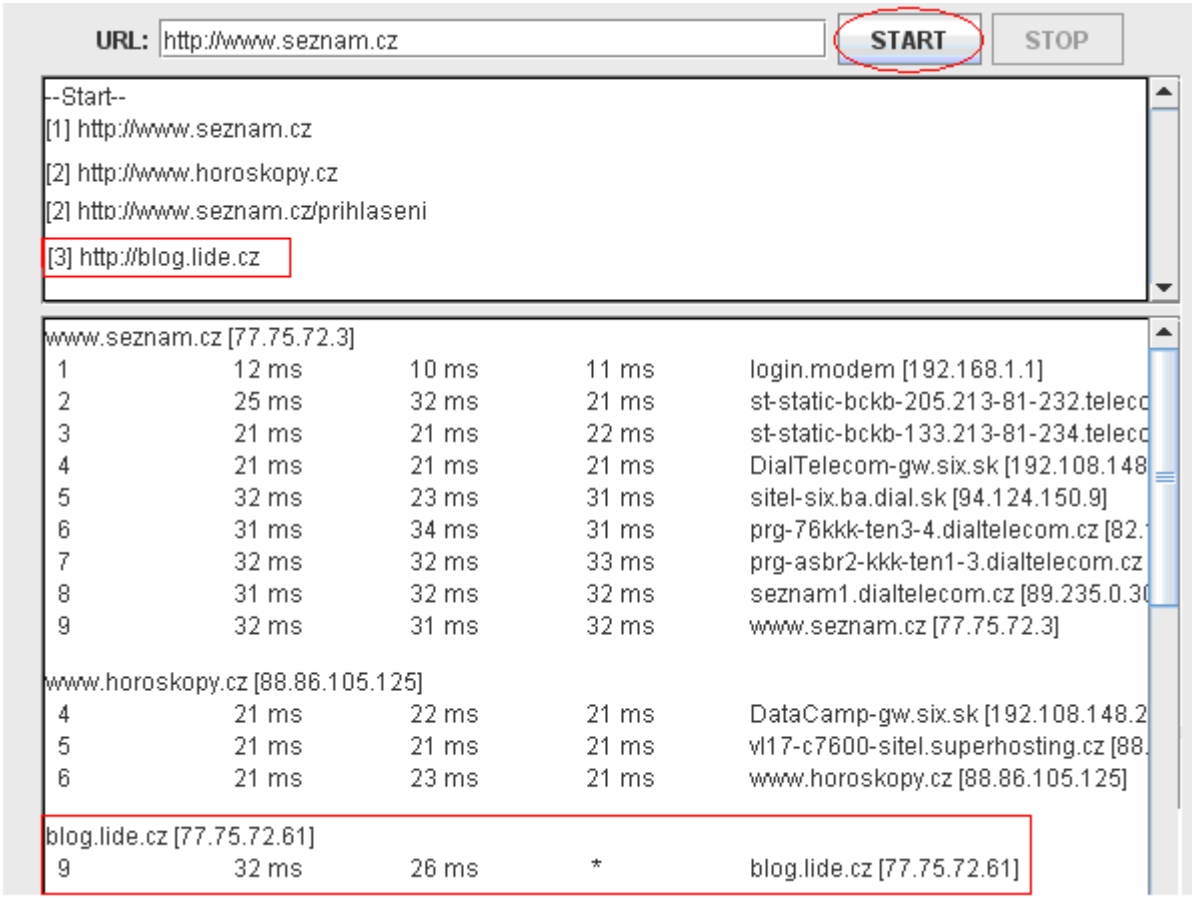
Obr. 37: Vytvorený robot pracujúci s protokolmi HTTP a ICMP

Môže nastať aj situácia, že sa v zachytených odkazech objaví odkaz, ktorý sa zhoduje s testovanou stránkou, tak na tento odkaz nebude vykonaný tracert. V našom prípade je použitá testovaná stránka www.seznam.cz a v zachytených odkazoch sa nachádza rovnaká stránka www.seznam.cz/prihlaseni, tak na túto stránku nebude vykonaný program tracert viz obr. 38.



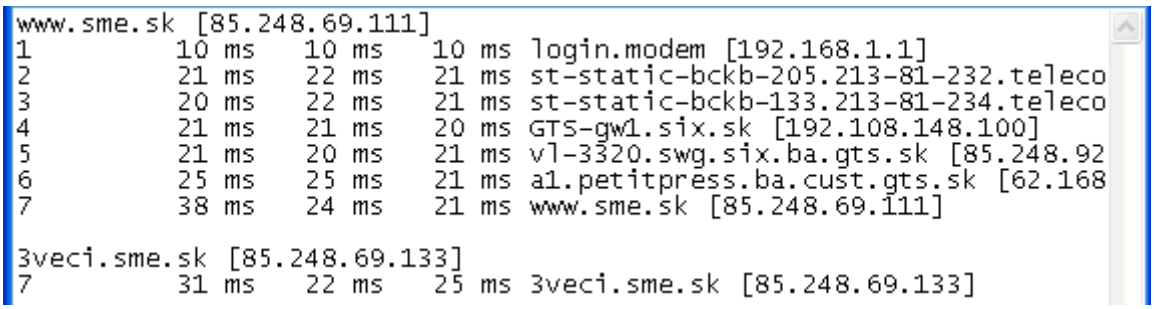
Obr. 38: Ukážka stránky, na ktorú nebude vykonaný program tracert

Ďalej môže nastať prípad, že sa vo výpise, ktorý vykonáva program tracert (spodná časť programu) objaví hviezdička (*), to znamená, že daný uzol (smerovač) neni dostupný. Pri nedostupnosti nejakého uzlu (smerovača) treba kontaktovať správcu siete. Ako testovaná stránka je použitá www.seznam.cz viz obr. 39.



Obr. 39: Ukážka nedostupnosi uzla (smerovača)

Na obr. 40 je znázornené ako sa výstup z aplikácie ukladá do súboru tracert.txt.



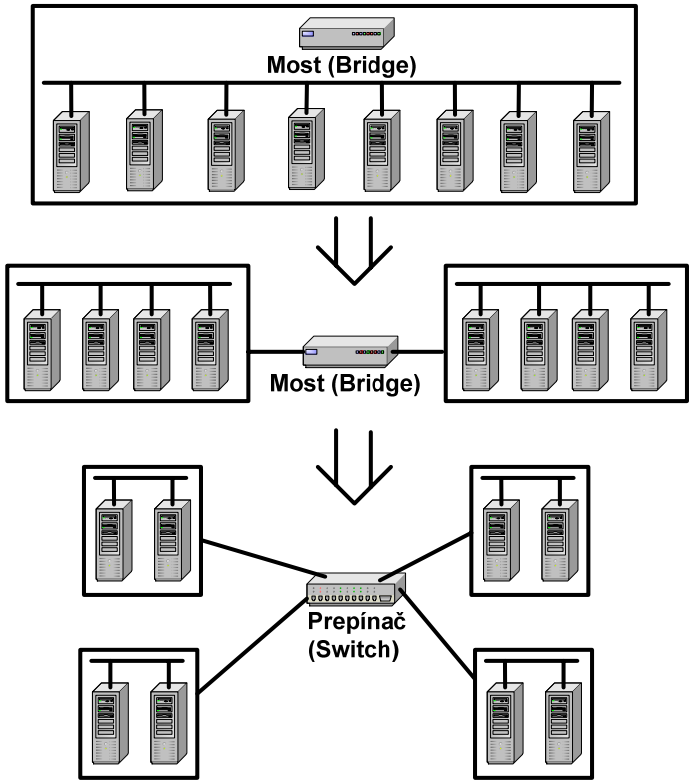
Obr. 40: Ukladanie výstupu programu do súboru tracert.txt

5. Priepustnosť (Throughput) siete

Priepustnosť je objem pridaných dát, ktoré môžu byť ešte prenesené sieťou za jednotku času, kedy sieť už obsahuje inú prevádzku [16]. Priepustnosť závisí na protokoloch použitých na prenos existujúcej prevádzky a novo pridanej prevádzky. Väčšina sieťovej prevádzky je v súčasnosti prenášaná ptokolom TCP.

5.1 Techniky pre zvýšenie priepustnosti siete: segmentácia

Jeden súvislý segment sa rozdelí na dve časti (dva segmenty) alebo na viac segmentov [17]. Čím budú čiastkové zdieľané segmenty menšie, tým menšia bude lokálna prevádzka a naopak bude väčšia prevádzka medzi čiastkovými segmentmi. Pri realizácií sú použité zariadenia most (bridge) a prepínač (switch) viz obr. 41.



Obr. 41: Techniky pre zvýšenie priepustnosti siete [17]

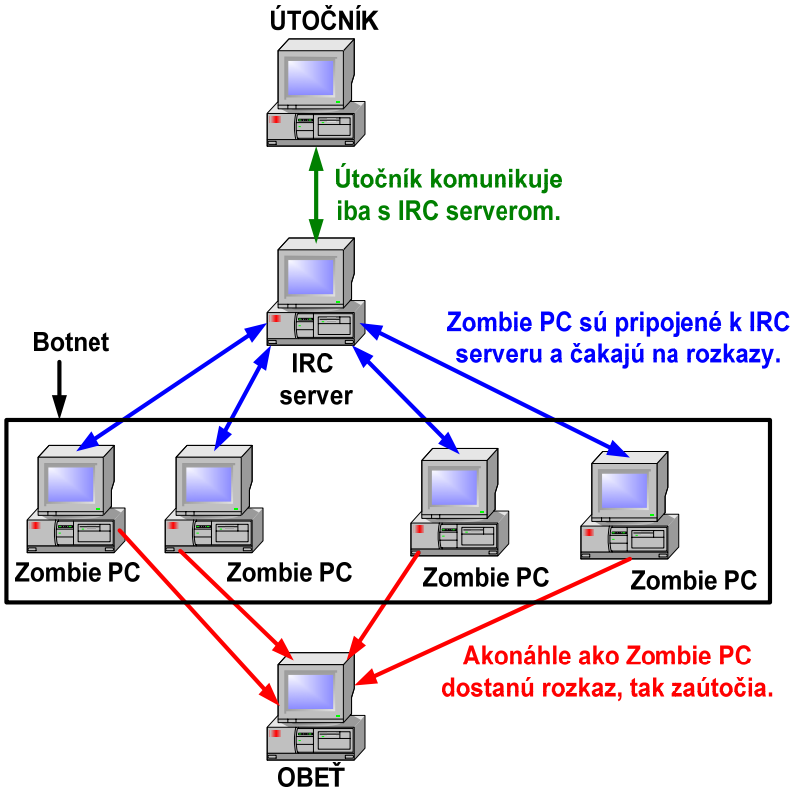
V nasledujúcom texte je vysvetlené čo znamená pojem most (bridge). Ďalšie zariadenie, ktoré je použité na obr. 35 je prepínač (switch). Toto zariadenie je popísané v kap. 2.2.1.

Most (Bridge) – Bridge znižuje veľkosť kolíznej domény, preto ho neni nutné konfigurovať. Pracuje na dátovej vrstve ISO/OSI modelu pričom zachytáva prichádzajúce dáta a rozhoduje sa, či ich má ďalej poslať alebo ich má vymazať. Bridge má podobnú funkciu ako switch, ktorý tiež operuje na druhej vrste [3].

6. DoS (Denial of Service) útoky

DoS útoky sú sieťové útoky, ktoré bránia prístupu ku službám. Tieto útoky blokujú aj služby siete zaplavovaním spojenia alebo bráni legitímnym klientom k prístupu ku službám siete. Tieto útoky môžu mať veľa podôb, od útoku jednoho paketu, ktoré spôsobujú zrútenie serveru až po koordinované záplavy paketov od mnoho hosťov a naopak pri záplavovom útoku sú zdroje na servery alebo na sieti narušené záplavou paketov. Pri napadnutý jednoho miesta ide záplaľavu ľahko identifikovať. Špecialnym prípadom DoS útoku je DDoS (Distributed Denial of Service). Útočník pri tomto útoku k zasiahnutiu cieľa používa veľké množstvo počítačov. Niektoré útoky majú napríklad jednoduchý plán ako poslať nekonečný prúd dát k zaplaveniu sieťového spojenia na servery ale môžu existovať aj útoky ako sú napríklad SYN záplavy, ktoré používajú upravené pakety k vyčerpaniu zdrojov za účelom zabrániť legitímnym klientom k pripojeniu na server [18].

Na obr. 42 je znázornený útok typu DDoS



Obr. 42: Útok typu DDoS [19]

Pomocou trojských koňov sa darí útočníkovi získať nové PC. Ak sa už nachádza tento vírus na počítači tak si stiahne potrebné programy a pripojí sa k IRC serveru, kde následne čaká na rozkazy. Takto sa postupne všetky nainfikované PC pripojá k IRC serveru (poznámka – každý útočník využíva iný server). Ak už je útočník pripojený k IRC serveru môže už všetkým počítačom rozkazovať. Útok je väčšinou prevedu tak, že útočník zadá príkaz, ktorý majú vykonať a na koho útočiť a zombie PC to splnia. Botnety sú siete zombie (infikovaných) počítačov. Tieto botnety môžu mať aj veľkosť stovky alebo tisíce počítačov [19].

6.1 Záplavové DoS útoky (DoS Flood)

Tieto útoky patria k najjednoduchším útokom. Ich filozófia spočíva v zahltení linky obete takým množstvom dát, že znemožnia regulárnu prevádzku. Ochrana proti ním je tažká a to z toho dôvodu ak sa útok prevádza z viacerých počítačov.

6.1.1 ICMP záplava (ICMP Flood)

Tento typ útoku používa protokol ICMP, najčastejšie sa používajú pakety typu ICMP Echo, čo sú pakety, ktoré využíva program ping. Pomocou tohoto programu zisťujeme dostupnosť vzdialeného zariadenia. Podľa doporučenia (RFC) by mala byť maximálna veľkosť ICMP Echo paketu 548 B, ale program ping pre systémy Linux a Windows umožňuje veľkosť ICMP Echo paketu až 65 kB (najväčší možný ICMP Echo paket môže byť 65 535 B, tak to uvádza špecifikácia). Podstata ICMP Echo je taká, že posielame ICMP Echo Request a cieľový počítač posiela späť ICMP Echo reply. Pritom zachováva veľkosť paketu. To ide využiť tak, že sfalšuje adresu odosielateľa a tým docielime, že dátová linka obete bude upchaná dvakrát. Raz dátami smerom tam a druhýkrát dátami späť (tieto dáta budú určené onej zfalšovanej adrese) [20].

6.1.2 Smurf Attack

Podobným útokom ako je útok ICMP Flood je útok smurf attack. Podstata tohoto útoku spočíva v tom, že útočník zahlcuje cieľový smerovač ICMP Echo paketmi, kde cieľová IP adresa je broadcastová adresa danej siete. Smerovač postupne zahltí sieť týmito broadcastovými správami. Týmto postupom môže útočník zahltiť aj sieť, z ktorej posiela dané pakety a to tak, že ako zdrojovú adresu ICMP paketov uvedie vlastnú sieť.

6.1.3 TCP záplavy (TCP Flood)

Tento typ útokov je založený na protokole TCP. Na Internete sa môžeme zoznámiť s pojmami ako sú napríklad SYN Flood, ACK Flood, RST Flood, FIN Flood, URG Flood, PSH Flood alebo ich kombináciami. Ich názvy vychádzajú z toho aké príznaky má TCP paket nastavené. Na tieto pakety obvykle počítač obete nereaguje. Správne by mal poslať späť TCP paket s RST príznakom ale toto nebýva rešpektované alebo je filtrované firewallom. Výnimkou sú pakety SYN a RST, ktoré patria do skupiny útokov prečerpávajúcich zdroje [20].

6.1.4 UDP záplavy (UDP Flood)

K tomuto útoku sa používa protokol UDP. Tento protokol je nespojovo orientovaný a pri prenose nie je potrebné nadviazať spojenie s cieľovou stanicou. Preto je zahlcovanie jednotlivými informačnými a potvrdzovacími správami nemožné. Používa sa tu aj zraniteľnosť služieb echo a chargen. Služba echo pracuje tak, že všetky dáta, ktoré prídu na jej port sú posielané späť a naopak služba chargen pracuje tak, že na ňu poielame nejaké dáta a ona vám späť posiela náhodné dáta. Trik spočíva v tom, že posielame dáta obeti na port echo a sfalšujeme zdrojovú adresu a port. Sfalšujeme ich tak, že zdrojovú adresu nastavíme na nejaký PC, ktorý poskytuje službu echo alebo chargen a port nastavíme na túto službu. Týmto docielime to, že

tieto dva počítače si budú posielať dáta stále dookola. Tieto služby sa v dnešnej dobe už nepoužívajú [20].

7. Systémy prevencie a narušenia (IPS a IDS)

Systémy pre odhaľovanie útokov a detekcia zraniteľnosti patria spolu s hraničnou sústavou firewall k neoddeliteľným prvkom zabezpečenia každého informačního systému.

Rad útokov sa odohráva na aplikačnej vrstve a používa techniky, ktoré firewall nedokáže odhaliť. To je úlohou IDS (Intrusion Detection System) a IPS (Intrusion Prevention System), ktoré predstavujú druhú obrannú líniu.

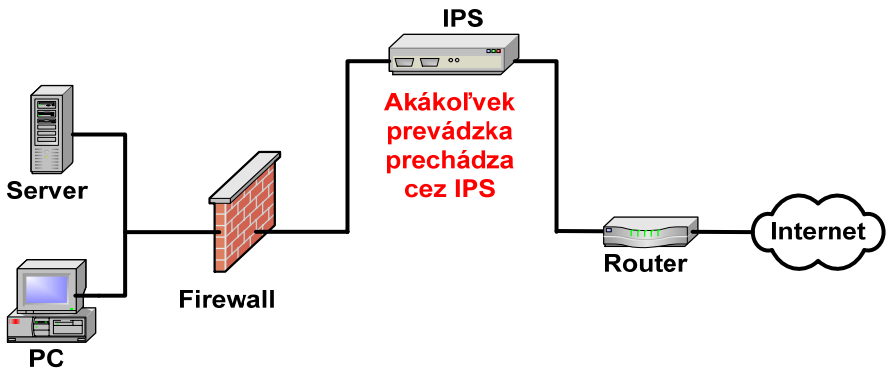
7.1 IPS a IDS systémy

Sieťové IPS a IDS sú zariadenia, ktoré sledujú sieťovú prevádzku. V tejto prevádzke hľadajú známky pokusov o prienik, neštandardné správanie klientov ale i šírenie nebezpečného softvéru (vírusov, trojských koní apod.).

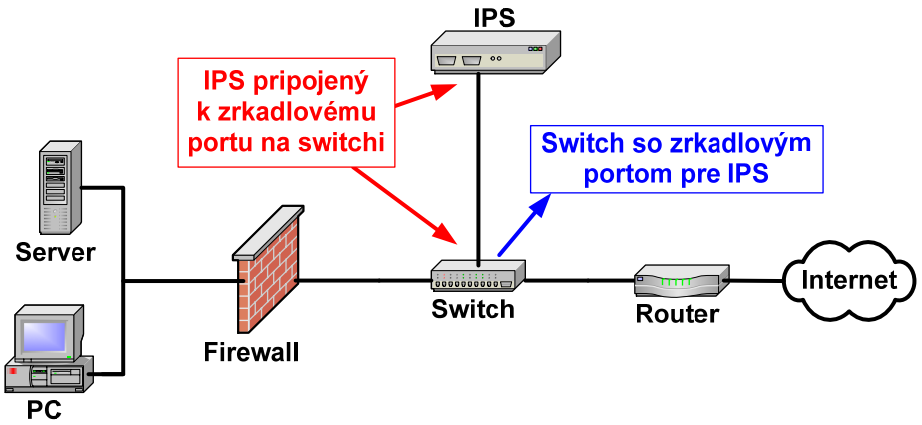
7.2 Systémy prevencie narušenia (IPS – Intrusion Prevention System)

B*lokujú škodlivú prevádzku a zabraňujú jej prístupu do systému, čím sa líšia od IDS, ktoré upozorňujú na ohrozenie, ktoré do systému už preniklo [21].* IPS vykonáva kompletnú inšpekciu paketov až po aplikačnú vrstvu a následne čistí internetovú prevádzku od vírusov, trójských koňov, apod. IPS tiež chráni aj vnútornú sieť pred útokmi typu DoS, DDoS.

Na obr. 43 a 44 je znázornený systém prevencie narušenia IPS.



Obr. 43: Systém prevencie narušenia IPS

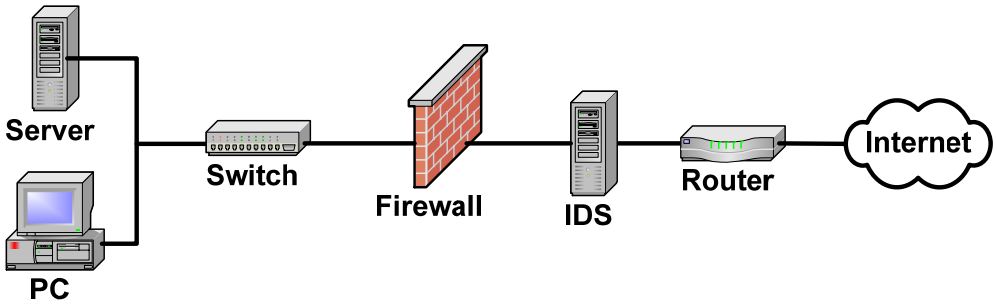


Obr. 44: Systém prevencie narušenia IPS

7.3 Systémy detekcie narušenia (IDS – Intrusion Detection System)

Jedná sa o riešenia slúžiace k detekcii pokusov o sieťové prieniky a útoky ako z vnútornej, tak z vonkajšej siete a následnou reakciou na narušenie [22]. IDS umožňujú stále sledovanie prevádzky ako na sieti, tak aj na serveroch. IDS sú na rozdiel od IPS pasivnými systémami, pretože podozrivú aktivitu iba zaznamenávajú, prípadne upozornia správcu siete zaslaním poplašnej správy. IDS ale nerobia žiadne opatrenia, ktoré by narušeniu zabránili.

Na obr. 43 je znázornený systém detekcie narušenia IDS.



Obr. 43: Systém detekcie narušenia IDS

8. Záverečné zhodnotenie

Diplomová práca je venovaná „Topologií sietí a ich monitorovaniu“. Táto práca je rozdelená an teoretickú a praktickú časť.

Teoretická časť je rozdelená na kapitoly. Prvá kapitola je venovaná základným rozdeleniam počítačových sietí: podľa rozlohy (LAN, MAN, WAN, PAN a stručne je popísaná technológia Bluetooth) a podľa topológie (fyzická a logická). Druhá kapitola je venovaná aplikačnej (siedmej) a sieťovej (tretej) vrstve ISO/OSI modelu a protokolom HTTP a ICMP. Piata kapitola je venovaná útokom na DNS a systémom smerovania paketov. Šiesta kapitola je venovaná DoS útokom a siedma kapitola je venovaná systémom prevencie a detekcie narušenia (IPS a IDS).

Piata kapitola je venovaná praktickej časti, ktorá je rozdelená na dve časti: prvá časť je venovaná návrhu a praktickej realizácií robota určeného k priezkumu a mapovaniu topológie počítačovej siete ľubovoľnej veľkosti na aplikačnej vrstve a to na protokole HTTP. Vytvorený robot dokáže na webovej stránky odkaz (link): vyhľadať, spracovať, vybrať a vypísať. Viac informácií o vytvorenom robotovi najdete v kap. 4.3. Druhá časť je venovaná návrhu a praktickej realizácií robota určeného k priezkumu a mapovaniu topológie počítačovej siete ľubovoľnej veľkosti na aplikačnej a sieťovej vrstve a to na protokole HTTP a ICMP. Pri tvorený tejto aplikácie sa využíva program z kap. 4.3, ktorý slúži na zachytávanie odkazov z webovej stránky a program tracert, ktorý vypisuje uzly (res. smerovače) na ceste paketov od zdrojového počítača k cieľovému počítaču. Vytvorený robot slúži tzv. mapovaniu ľubovoľnej počítačovej siete. Viac informácií o vytvorenom robotovi najdete v kap. 4.4. Ďalej mal byť realizovaný robot, ktorý bude analyzovať priepustnosť jednotlivých segmentov siete, ale po dohode s vedúcim diplomej práce bola táto časť spracovaná iba teoreticky. Vytvorená aplikácia je realizovaná v programovacom jazyku Jáva s vývojovým prostredím NetBeans 6.8.

Všetky potrebné informácie k diplomovej práci (elektronický text diplomovej práce, zdrojový kód vytvorenej aplikácie) sú k dispozícií v priloženom CD.

Použitá literatúra

[1] POPPE, V. *Úvod do počítačových sietí.* Žilina 2004 [online]. [cit. 2010-04-21]. Dostupné z WWW: <http://fria.fri.uniza.sk/~vapo/vyuka/poc-siet.htm>.

[2] *Rozdelenie sietí.* [online]. [cit. 2010-04-21]. Dostupné z WWW: <http://www.gt12.sk/predmety/inf/materialy/ucebnica/pocitacove_systemy/pocitacove_systemy.htm>.

[3] PALÁSTHY, J. *Technickéj prostriedky pre vytvorenie sietí.* [online]. [cit. 2010-04-21]. Dostupné z WWW: <http://www.oskole.sk/?id_cat=1008&clanok=1859>.

[4] JEŘÁBEK, J. *Pokročilé komunikační techniky.* Brno 2010 [cit. 2010-04-21].

[5] *HTTP protokol.* [online]. [cit. 2010-04-21]. Dostupné z WWW: <http://neuron-ai.tuke.sk/~hudecm/Tutorials/HTTP%20Protokol.html>.

[6] DOSTÁLEK, L., KABELOVÁ, A. *Velký průvodce protokoly TCP/IP a systémem DNS.* 3. vyd. Praha: Computer Press, 2002. 542 s. ISBN 80-7226-675-6.

[7] BURDA, K. *Síťová vrstva počítačových sítí-protokoly.* Brno 2010 [cit. 2010-04-21].

[8] PETERKA, J. *IP smeřování.* Praha 2008 [online]. [cit. 2010-04-21]. Dostupné z WWW: <http://www.earchiv.cz/l219/nahled.php3?l=6&me=1>.

[9] DNS (Domain Name System). 2008 [online]. [cit. 2010-04-21]. Dostupné z WWW: <http://deja-vix.sk/sysadmin/dns.html>.

[10] SCAMBRAY, J., MCCLURE, S., KURTZ, G. *Hacking bez tajemství*. Praha: Computer Press, 2001. 592 s. ISBN 80-7226-549-0.

[11] ALLEN H., SHON H., CHRIS E., JONATHAN N., MICHAEL L. *Hacking – manuál hackera*. Grada Publishing a.s., 2008. 399s. ISBN 8024713462, 9788024713465

[12] ESET : Uživatelská příručka – *Typy útoků.* [online]. [cit. 2010-04-21]. Dostupné z WWW: <http://download.eset.com/manuals/ESET_ESS_User_Guide_CZ.PDF>.

[13] Sun Developer Network (SDN), The Source for Java Developers. Sun Microsystems, Inc. 1994 – 2009 [online]. [cit. 2009-12-15]. Dostupné z WWW. <http://java.sun.com/>.

[14] The Java *TM* Tutorials, How to Use the System Tray. Sun Microsy 1995 – 2009 [online]. [cit. 2009-12-15]. Dostupné z WWW: <http://java.sun.com/docs/books/tutorial/>.

[15] Netbeans Docs & Support, Documentation, Training and Support [online]. [cit. 2009-12-15]. Dostupné z WWW: <http://www.netbeans.org/kb>.

[16] CESNET : *Sledovaní a optimalizace výkonnostních charakteristik.* [online]. [cit. 2010-04-21]. Dostupné z WWW: <http://www.cesnet.cz/doc/2008/zprava/e2eperf.html>.

[17] PETERKA, J. *Internetworking –II.* Praha 2008 [online]. [cit. 2010-04-21]. Dostupné z WWW: <http://www.earchiv.cz/l218/nahled.php3?l=14&me=1>.

[18] CompuNet : *DoS a DDoS útoky a ochrana.* [online]. [cit. 2010-04-21]. Dostupné z WWW: <http://www.compunet.cz/bezpecnost-siti-dos.php>.

[19] HALLER, M. *Denial of Service útoky: distribuované DoS.* 2006 [online]. [cit. 2010-04-21]. Dostupné z WWW:

http://www.lupa.cz/clanky/denial-of-service-utoky-vyuziti-mitm-utoku/

[20] HALLER, M. *Denial of Service (DoS) útoky: záplavové typy.* 2006 [online]. [cit. 2010-04-21]. Dostupné z WWW: < http://www.lupa.cz/clanky/denial-of-service-dos-utoky-zaplavove-typy/ >.

[21] Actinet : *Systémy prevencie narušenia (IPS - Intrusion Prevention Systems).* [online]. [cit. 2010-04-21]. Dostupné z WWW: <http://www.actinet.sk/bezpecnost_informacnich_technologii/d23/l2/m9/j1/s29/System_security.html>.

[22] Actinet : *Systémy detekcie narušenia (IDS – Intrusion Detection Systems).* [online]. [cit. 2010-04-21]. Dostupné z WWW: <http://www.actinet.sk/bezpecnost_informacnich_technologii/d23/l2/m9/j1/s29/System_security.html>.

Zoznam skratiek

LAN	–	Local Area Network
MAN	–	Metropolitan Area Network
WAN	–	Wide Area Network
IrDA	–	Infrared Data Association
MAC	–	Media Access Control
HTTPS	–	Hypertext Transfer Protocol Secure
SMTP	–	Simple Mail Transfer Protocol
HTTP	–	Hypertext Transfer Protocol verzie
HTTP/0.9	–	Hypertext Transfer Protocol verzie 0.9
HTTP/1.0	–	Hypertext Transfer Protocol verzie 1.0
HTTP/1.1	–	Hypertext Transfer Protocol verzie 1.1
WWW	–	World Wide Web
IP	–	Internet Protocol
IPv4	–	Internet Protocol verzie 4
IPv6	–	Internet Protocol verzie 6
IGMP	–	Internet Group Management Protocol
OSPF	–	Open Shortest Path First
IGRP	–	Interior Gateway Routing Protocol
EIGRP	–	Enhanced Interior Gateway Routing Protocol
IPsec	–	IP security
ICMP	–	Internet Control Message Protocol
TTL	–	Time To Live
RTT	–	Round Trip Time
UDP	–	User Datagram Protocol
RIP	–	Routing Information Protocol
RIPv1	–	Routing Information Protocol verzie 1
RARP	–	Reverse Address Resolution Protocol
DNS	–	Domain Name System
DNSSEC	–	DNS Security Extensions
JVM	–	Java Virtual Machine
URL	–	Uniform Resource Locator

LIFO	–	Last Input First Output
DoS	–	Denial of Service
DDoS	–	Distributed Denial of Service
IDS	–	Intrusion Detection System
IPS	–	Intrusion Prevention System

Zoznam príloh

Prvá príloha

V prvej prílohe, ktorá je uvedená na CD sa nachádza elektronický text diplomovej práce.

Druhá príloha

V druhej prílohe, ktorá je uvedená na CD sa nachádza vytvorený robot, ktorý je určený k prieskumu a mapovaniu topológie počítačovej siete ľubovolnej veľkosti a to na protokole HTTP a ICMP.